

Escuela Politécnica Superior

19  
20

# Trabajo fin de grado

Aplicación móvil de turismo multi-patrimonio



William Guerrero

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente nº 11



**UNIVERSIDAD AUTÓNOMA DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Aplicación móvil de turismo multi-patrimonio**

**Autor: William Guerrero**

**Tutor: Iván Cantador**

**julio 2020**

**Todos los derechos reservados.**

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

**DERECHOS RESERVADOS**

© 3 de Julio de 2020 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, nº 1

Madrid, 28049

Spain

**William Guerrero**

*Aplicación móvil de turismo multi-patrimonio*

**William Guerrero**

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN



*A todas las personas que hicieron de la carrera la mejor época de mi vida*

*Para el logro del triunfo siempre ha sido indispensable pasar por la senda de los sacrificios.*

*Simón Bolívar*



# PREFACIO

---

Siento gran cariño por este proyecto ya que une el lugar de donde vengo con el lugar donde vivo, así que cuando se me propuso, sentí un gran interés por la idea y el concepto, de tal manera que este no es solo un proyecto de TFG, sino una unión entre culturas que me representan.

William Guerrero



# AGRADECIMIENTOS

---

En primer lugar quiero agradecer a mi familia, que aunque están lejos siempre me han apoyado en todo momento y se han sentido orgullosos de los logros que he conseguido hasta ahora.

Agradecer además a todas las personas especiales que he conocido durante la carrera y que han aportado mucho a mi vida, además de ser un apoyo siempre que necesitas segundas opiniones y ellos están ahí.

Por supuesto también, las gracias a mi tutor Iván Cantador por acompañarme en el desarrollo del TFG e irme guiando a través del recorrido con mucha paciencia y con gran dedicación.

Un agradecimiento especial a Fernando, quien fue la persona que más me apoyó a entrar en la carrera y me dio la fuerza para continuar cuando el camino parecía difícil.

Además también a un gran amigo llamado Jose Alejandro quien me recomendó muchas de las cosas que luego pude utilizar para este TFG, por lo que le estoy muy agradecido.

Y por último a agradecer a Alejandra quien me apoyó día tras día a seguir con el TFG, por ayudarme cuando necesitaba un consejo o no sabía por donde continuar, siempre ha estado ahí.

Este trabajo se ha desarrollado en el marco del proyecto “América en Madrid. Patrimonios interconectados e impacto turístico en la Comunidad de Madrid” (AmerMad-CM), financiado por la Comunidad de Madrid (ref. H2019/HUM-5694).



# RESUMEN

---

En este trabajo de final de grado, se describe el proceso de diseño y construcción de una aplicación turística destinada a ser altamente escalable y reutilizable en el futuro.

Una de las peculiaridades de dicho desarrollo es el uso de dos frameworks, a saber, el SDK de Android (Software Development Kit) y Spring Framework.

La aplicación, llamada **MadLat**, será de gran ayuda para vincular dos comunidades con una gran historia, ya que interconecta el conocimiento de diversas formas de patrimonio cultural y natural entre América Latina y la Comunidad de Madrid.

Así mismo, la aplicación se puede extender a otras plataformas además de Android, para tal fin, la aplicación se divide en dos componentes: un primer componente construido sobre Android que es visual y conlleva un bajo costo de cómputo, y un segundo componente que implica un mayor costo computacional y se implementa en servidores que usan Spring Framework.

Por último, MadLat sigue una arquitectura basada en microservicios que tiene como objetivo modularizar toda la lógica tanto como sea posible, para facilitar la integración futura de nuevos módulos. Por lo tanto, los módulos anteriores no deben conocerse o pueden implementarse en diferentes lenguajes de programación o utilizar otros frameworks.

# PALABRAS CLAVE

---

América latina, Madrid, Spring Framework, Android, Arquitectura microservicios





# ABSTRACT

---

In this degree final project, it is described the process of designing and building a tourism application aimed to be highly scalable and reusable in the future.

One of the peculiarities of such development is the use of two frameworks, namely the Android SDK (Software Development Kit) and the Spring Framework.

The application, called **MadLat**, will be of great help for linking two communities with a large history. It interconnects knowledge from various forms of cultural and nature heritage between Latin America and the Community of Madrid.

Also, the application can be extended to other platforms besides Android. For such purpose, the application is divided into two components: a first component built upon Android that is visual and entails low computation cost, and a second component that entails more computational cost and is deployed in servers that use the Spring Framework.

Lastly, MadLat follows a micro-service based architecture that aims to modularize all the logic as much as possible, in order to make the future integration of new modules much easier. Hence, previous modules have not to be known or may be implemented in different programming languages or use other frameworks.

# KEYWORDS

---

Latin America, Madrid, Spring Framework, Android, Microservices architecture



# ÍNDICE

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación .....	1
1.2	Objetivos .....	1
1.3	Organización de la memoria .....	2
1.4	Diagrama de Gantt .....	2
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Estudio de aplicaciones relacionadas .....	5
2.1.1	ARAMEDPI .....	5
2.1.2	Barcelona Art & Culture .....	6
2.1.3	Ciudades patrimonio de la humanidad .....	8
2.1.4	World Heritage .....	9
2.1.5	Culture Trip .....	11
2.1.6	UNESCO Heritage .....	14
2.2	Comparativa .....	15
2.2.1	Comparativa de elementos funcionales .....	15
2.2.2	Comparativa de elementos audiovisuales .....	16
<b>3</b>	<b>Análisis</b>	<b>19</b>
3.1	Requisitos funcionales .....	19
3.2	Requisitos no funcionales .....	20
3.3	Casos de uso .....	20
3.3.1	Actores en el sistema .....	20
3.3.2	Diagrama de casos de uso .....	21
3.3.3	Descripción de casos de uso .....	22
<b>4</b>	<b>Diseño</b>	<b>31</b>
4.1	Prototipo de la aplicación .....	31
4.1.1	Logotipo de la aplicación .....	31
4.1.2	Prototipo de la interfaz .....	32
4.2	Arquitectura de la aplicación .....	32
4.3	Modelo de datos .....	33
4.4	Diagramas entidad relación y base de datos .....	33
<b>5</b>	<b>Desarrollo</b>	<b>37</b>

5.1 Herramientas y entornos .....	37
5.2 Librerías .....	37
5.3 Microservidores .....	38
5.3.1 Acceso a base de datos .....	39
5.3.2 Creación de API REST .....	39
5.4 Android .....	42
5.5 Resultados .....	42
<b>6 Conclusiones y trabajo futuro</b>	<b>47</b>
6.1 Conclusiones .....	47
6.2 Trabajo futuro .....	48
<b>7 Anexo</b>	<b>49</b>
<b>Bibliografía</b>	<b>54</b>

# LISTAS

---

## Lista de figuras

1.1	Diagrama de Gantt TFG .....	3
2.1	Vistas de la interfaz de ARAMEDPI .....	6
2.2	Vistas de la interfaz de Barcelona Ar Culture .....	7
2.3	Vista del mapa la aplicación Barcelona Art & Culture .....	8
2.4	Interfaz de la aplicación Ciudades patrimonio de la humanidad .....	9
2.5	Interfaz de la aplicación Heritage .....	10
2.6	Menú lateral de la aplicación Worl Heritage .....	11
2.7	Vista principal de la aplicación Culture Trip .....	12
2.8	Interfaz de la aplicación Culture Trip .....	13
2.9	Interfaz de la aplicación World Heritage .....	14
3.1	Diagrama de casos de uso .....	21
4.1	Logotipo de la aplicación .....	31
4.2	Arquitectura del sistema .....	34
4.3	Modelos de datos .....	35
4.4	Diagrama E-R .....	35
5.1	Uso de Spring Boot en la clase principal .....	38
5.2	Uso de Spring Boot en la clase principal con configuración .....	38
5.3	Uso de Spring JPA .....	39
5.4	Creación de tablas desde Java .....	40
5.5	Archivo de configuración de spring .....	41
5.6	API REST con Spring .....	41
5.7	Creación de endpoints de la API REST .....	42
5.8	Llamada con la API REST .....	42
5.9	Listas de puntos de la aplicación .....	43
5.10	Llamada completa desde el front hasta el back .....	44
5.11	Vistas de la aplicación .....	45
7.1	Información de un elemento .....	49
7.2	Vista de favoritos de la aplicación .....	50
7.3	Vista inicial de la aplicación .....	50

7.4	Vista del inicio de sesión de la aplicación .....	51
7.5	Vista del mapa en la aplicación .....	51
7.6	Vista del perfil del usuario en la aplicación .....	52

## Lista de tablas

2.1	Comparativa de elementos funcionales .....	15
2.2	Comparativa de elementos audiovisuales de otras aplicaciones .....	16
3.1	Caso de uso CU-1 .....	22
3.2	Caso de uso CU-2 .....	23
3.3	Caso de uso CU-3 .....	23
3.4	Caso de uso CU-4 .....	24
3.5	Caso de uso CU-5 .....	24
3.6	Caso de uso CU-6 .....	25
3.7	Caso de uso CU-7 .....	25
3.8	Caso de uso CU-8 .....	26
3.9	Caso de uso CU-9 .....	26
3.10	Caso de uso CU-10 .....	27
3.11	Caso de uso CU-11 .....	27
3.12	Caso de uso CU-12 .....	28
3.13	Caso de uso CU-13 .....	28
3.14	Caso de uso CU-14 .....	29

# INTRODUCCIÓN

---

En este capítulo se analizarán los porqués de este proyecto y como se estructura el documento, también se encuentra la organización temporal que se ha seguido para la realización del proyecto.

## 1.1. Motivación

A través de este proyecto se busca dar a conocer los datos sobre un pasado y un presente compartido entre la Comunidad de Madrid y América Latina.

Existe poca información sobre la llegada de americanos y europeos retornados, desde el continente americano, hacia el viejo continente, por lo que se busca también visibilizar esa parte de la historia compartida del movimiento entre ambos pueblos.

Siendo además perteneciente a la comunidad latina en Madrid con ascendencia europea, el hecho de conectar estos dos mundos es parte importante no solo de la cultura sino de mis propias motivaciones personales y mi identidad.

## 1.2. Objetivos

Entre los objetivos de este proyecto están:

- Dar a conocer ese patrimonio interconectado del que hemos hablado y que a través de esta aplicación intente aumentar su visibilidad entre la población.
- Crear una aplicación Android que cumpla con el objetivo de la visibilidad.
- Que la aplicación a crear pueda reutilizarse en parte para otras plataformas en un futuro como iOS o una versión web.
- Aumentar los conocimientos en Android, Java y Spring, así como aumentar el dominio en el uso de distintos frameworks.
- Por último profundizar en una arquitectura reciente, que otorgue a la aplicación las facilidades de nuevos desarrollos futuros dentro del mismo marco de la aplicación actual.

## 1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Estado del arte:** En este capítulo se analiza el estado actual de las aplicaciones que existen y realizan una actividad similar a la que se busca, tanto para exponer patrimonios culturales, como para incentivar el turismo en la región.
- **Análisis:** Se corresponde con una fase de análisis del proyecto en la que se define todo lo que este debe tener, inspirándose además en los elementos que ya existen y se vieron en el capítulo anterior.
- **Diseño:** En este apartado se establece como debe ser el sistema a nivel de interfaz, arquitectura del sistema y a nivel lógico.
- **Desarrollo:** En este apartado se ven las herramientas utilizadas y aprendidas durante la fase de desarrollo que han sido necesarias para poder llevar a cabo el proyecto, así como algunas capturas de los resultados obtenidos.
- **Conclusiones y trabajo futuro:** Por último en este capítulo, se observa como debe ser el trabajo futuro después de finalizado el proyecto si a este se deciden añadir otras funcionalidades y las conclusiones que se han sacado.

## 1.4. Diagrama de Gantt

El diagrama de Gantt con la planificación del TFG puede verse en la figura 1.1.



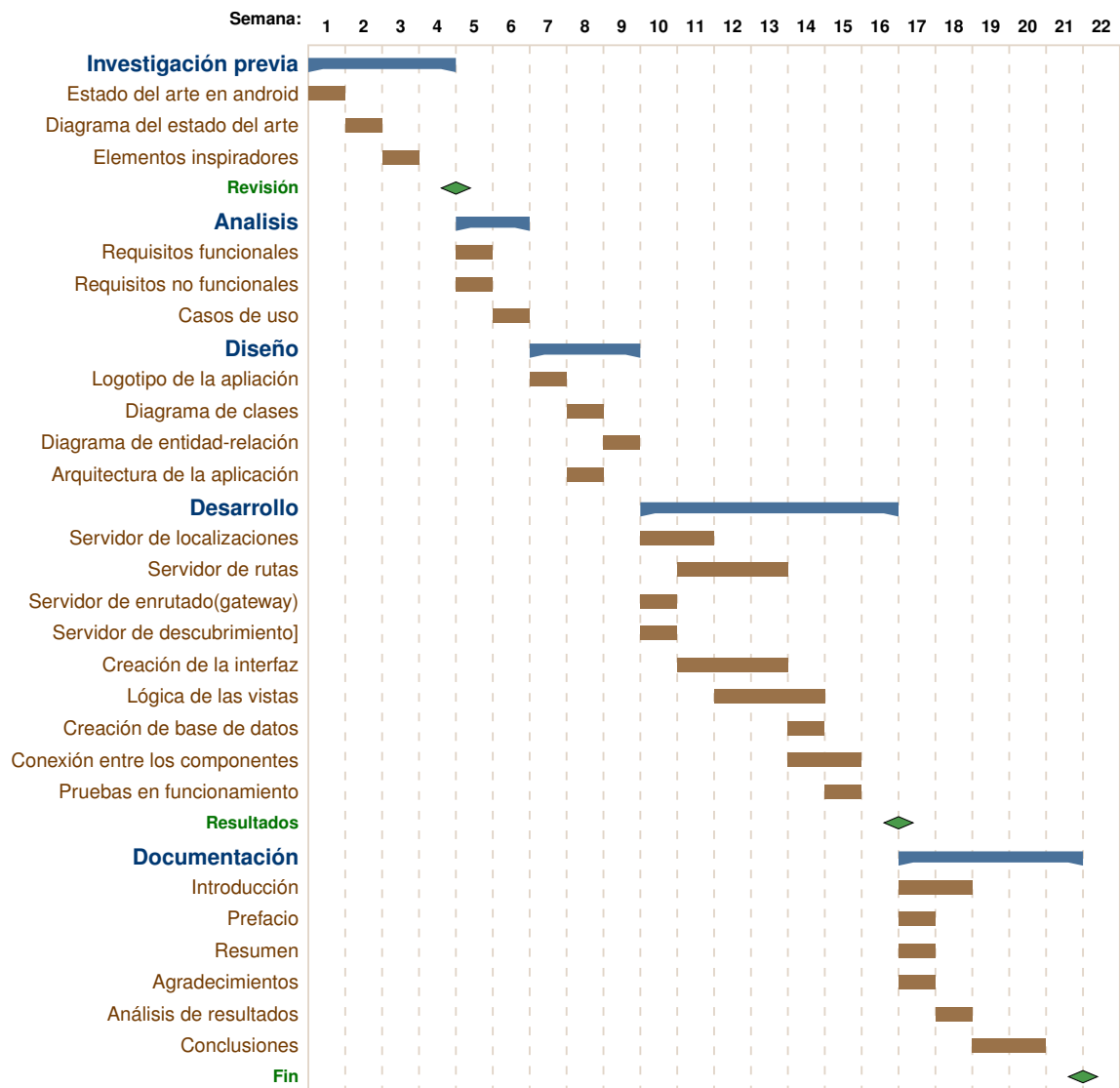


Figura 1.1: Diagrama de Gantt con la planificación del TFG.



## ESTADO DEL ARTE

---

### 2.1. Estudio de aplicaciones relacionadas

En este capítulo se lleva a cabo un estudio sobre aplicaciones de cultura y arte que estén actualmente en el mercado.

En dicho estudio se buscan aplicaciones para dispositivos Android que estén relacionadas con la cultura iberoamericana y con contenido que lleve información asociada, como imágenes, descripciones, mapas, entre otros datos. En su mayoría estas tienen imágenes con explicaciones de las piezas, ya sean pinturas, edificios, estatuas o personajes históricos.

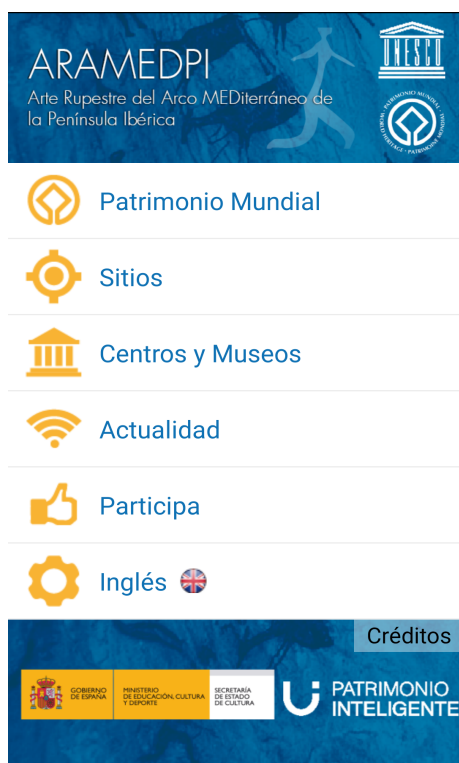
#### 2.1.1. ARAMEDPI

Arte rupestre del arco Mediterráneo de la Península Ibérica o también conocido como ARAMEDPI [1] es una aplicación que se utiliza para explorar el arte rupestre repartido por 6 comunidades autónomas de España, permite ver los sitios, centros y museos donde se encuentran estas obras y dar un paseo virtual a través de imágenes y descripciones para conocer un poco más sobre el arte rupestre.

Esta aplicación dispone solamente de un menú principal como se ve en la figura 2.1(a) y en la que se encuentran las siguientes 6 opciones:

- **Patrimonio Mundial:** Se muestra un resumen del recorrido y la historia organizativa que tuvo la comisión encargada de desarrollar el plan de actuación sobre el arte rupestre en 6 comunidades autónomas.
- **Sitios:** Se ve una lista de todos los sitios donde se ha encontrado arte rupestre en la península, es posible acceder en cada uno de los elementos y ver una descripción e imágenes sobre el sitio en cuestión.
- **Centros y Museos:** Al igual que el anterior se encuentra una lista, en este caso de todos los centros y museos, igualmente cada uno con una descripción e imágenes.
- **Actualidad:** Actualmente no funciona, pero se entiende como un apartado para leer novedades y noticias sobre el arte rupestre y los sitios donde verlo.
- **Participa:** Se muestran una serie de botones para compartir la aplicación en redes sociales con el dibujo con cada una de estas redes sobre el botón.

- **Idioma:** Este botón cambia completamente los textos de toda la aplicación entre inglés o español, no tiene ningún submenú.



(a) Menú principal de la aplicación ARAMEDPI



(b) Elemento mínimo ARAMEDPI

**Figura 2.1:** Vistas de la aplicación ARAMEDPI

Se puede destacar que en los elementos de Sitios, Centros y Museos además de la descripción y las imágenes se dispone de algunas opciones extras en un menú superior como se ve en la figura 2.1(b), que son: Como llegar, Tipo de arte, Horario de atención al público, página web, correo electrónico y teléfono.

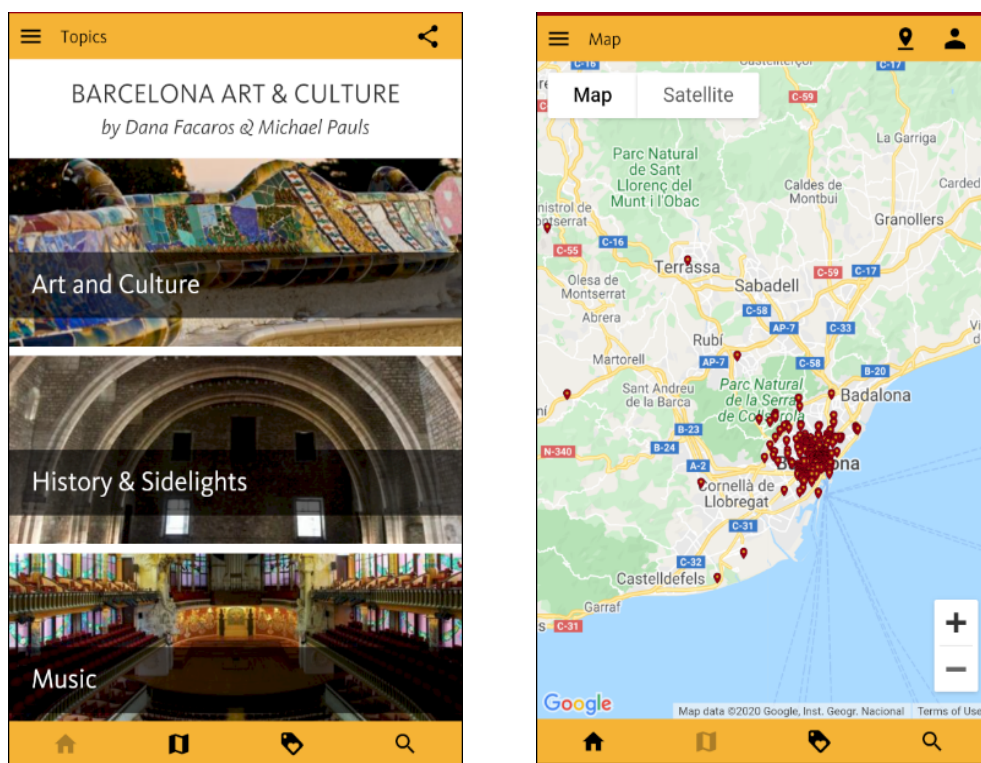
Por último, existe la opción de visualizar el menú inicial en cualquier momento pulsando en un botón de menú en el lado superior derecho.

### 2.1.2. Barcelona Art & Culture

La aplicación de Barcelona Art & Culture [2] sirve para localizar lugares de interés de la ciudad de Barcelona, leer sobre su historia y consultar los eventos más importantes de la ciudad.

Al entrar en la aplicación hay una lista donde se observan las categorías en las que se pueden navegar como se ve en la figura 2.2(a), las cuales son: Art and Culture, History & Sidelights, Music, Nightlife & Performance, People, Places, Practical Info y Top Sights. Además de estas categorías, dentro de cada una se encuentran subcategorías temáticas, una vez es seleccionado un elemento, estos,

como se ve en la figura 2.3, tienen dentro una descripción bastante completa con enlaces externos y algunas imágenes de referencia.



(a) Vista inicial

(b) Vista del mapa

**Figura 2.2:** Distintas vistas de la aplicación Barcelona Art & Culture transicionando con el menú inferior.

En la parte inferior como se ve en la figura 2.2(a) se dispone de una barra de menú fija, en la que es posible encontrar el inicio, un mapa y ver los sitios y eventos de la ciudad, existe también una sección de favoritos en la que los usuarios pueden guardar los elementos de la aplicación, ya sean, actividades, lugares, eventos o consejos de viaje y por último un botón de búsqueda para encontrar rápidamente un elemento dentro de la aplicación por su nombre.

En la configuración de la aplicación es posible cambiar el tamaño de la fuente, leer acerca de la aplicación, comprar la versión sin anuncios y ver los sitios que se encuentran cercanos.

Uno de los aspectos relevantes de la aplicación mencionados anteriormente es el mapa como muestra la figura 2.2(b), el acceso a los elementos desde el mapa es mucho más conveniente para el usuario a la hora de ver los sitios que tiene cerca o a la hora de elaborar una ruta turística.



**Figura 2.3:** Vista del mapa la aplicación Barcelona Art & Culture

### 2.1.3. Ciudades patrimonio de la humanidad

Esta aplicación [3] tiene como finalidad que los usuarios realicen recorridos en los que, con imágenes y audio, se accede a los diferentes lugares patrimonio de la humanidad de España.

Al entrar en la aplicación se visualiza un mapa que muestra los diferentes lugares en donde se encuentran ubicados los diferentes patrimonios de la humanidad ubicados en España, además dispone de varios botones para comenzar a navegar en el que existe una presentación en vídeo de la aplicación, un botón de información que muestra otras aplicaciones de temática similar y el botón principal de visita guiada.

Dentro de la visita es posible escoger por temas, distribuidos en bocadillos que se desplazan horizontalmente y en cada tema se puede seleccionar si se quiere algún añadido, audio-descripción, subtítulos y lengua de signos; dentro de la visita guiada también se reproducen una serie de vídeos que explican a los usuarios sobre lugares que corresponden con la temática escogida.

Una de las ventajas de esta aplicación es que estos recorridos se pueden descargar y tener el contenido disponible aunque no se este conectado a ninguna red como se ve en la figura 2.4, además antes de descargar se conoce el tamaño que ocupará en el dispositivo y de esta forma también saber cuanto consumirá aproximadamente si se utiliza una tarifa de datos.

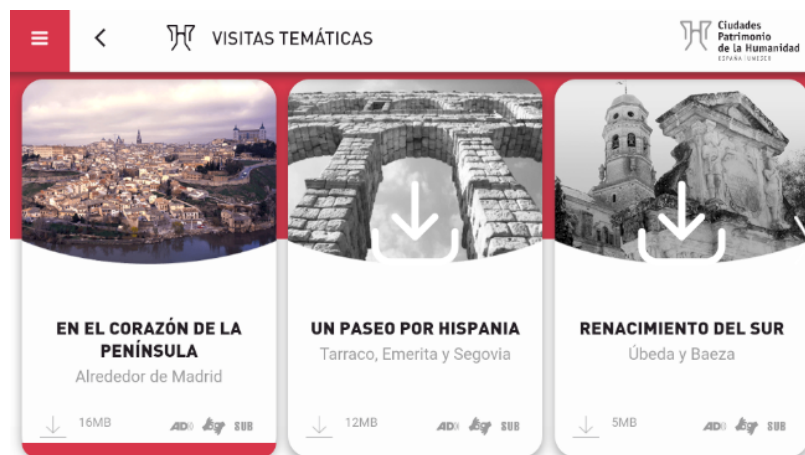


Figura 2.4: Vista de las descargas disponibles

### 2.1.4. World Heritage

Con esta aplicación [4] es posible encontrar los sitios de cada país que son patrimonio de la humanidad e incluso los que son compartidos, como por ejemplo algunos patrimonios naturales.

Al entrar en la aplicación como se ve en la figura 2.5(a), no tiene menú, sino una lista donde se puede explorar en vertical y en orden alfabético por países los patrimonios que existen, estos se presentan como pequeñas tarjetas con una imagen y el nombre en la parte de abajo, también tienen un botón con forma de corazón para guardarlo como favorito, la forma en la que se entra y se visualiza la información de cada patrimonio es pulsando en la imagen del lugar; en la parte superior de la pantalla se encuentra una lupa con la que se busca entre los distintos patrimonios.

Dentro de cada patrimonio hay una descripción y algunas fotos del lugar de interés pulsando en un menú superior que posee como se ve en la figura 2.5(b).

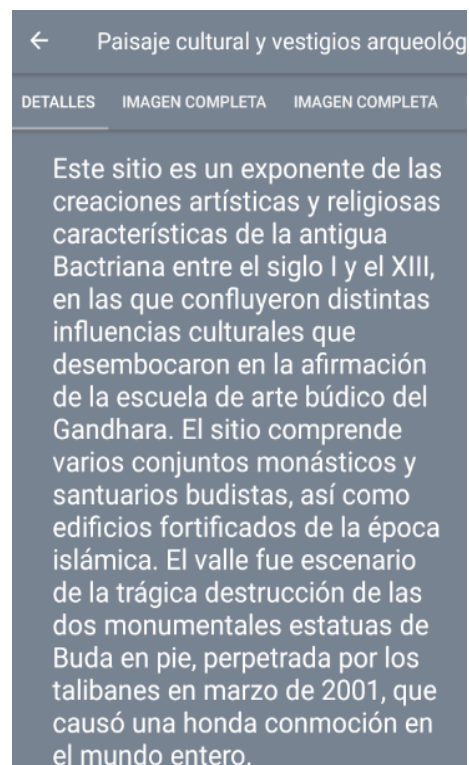
Si desde la vista inicial se selecciona en el botón superior izquierdo de menú, aparecerá el menú que aparece en la figura 2.6 con las opciones que se describen a continuación:

- **Tipo:** Pulsando este botón se accede a la vista inicial en la que se ven todos los patrimonios.
- **Lista:** Realiza la misma acción que el botón anterior.
- **Le gusta:** En este apartado se encuentran los elementos que hemos marcado como favoritos con el botón de corazón.
- **Idioma:** La aplicación dispone de hasta 11 idiomas diferentes que se pueden cambiar pulsando sobre el nombre del idioma.
- **Calificación:** Esta opción lleva al usuario a la página de Google Play donde es posible que este evalúe la aplicación de 0 a 5 estrellas.
- **Ver logros:** La aplicación se conecta con Google Play Games y sincroniza logros por eventos que se realizan en la aplicación, en este apartado se encuentran los logros que se han conseguido hasta ahora.
- **Mapa:** Enseña una vista de mapa desde satélite en la que se exploran las ubicaciones de los patrimonios de la humanidad.





(a) Vista inicial

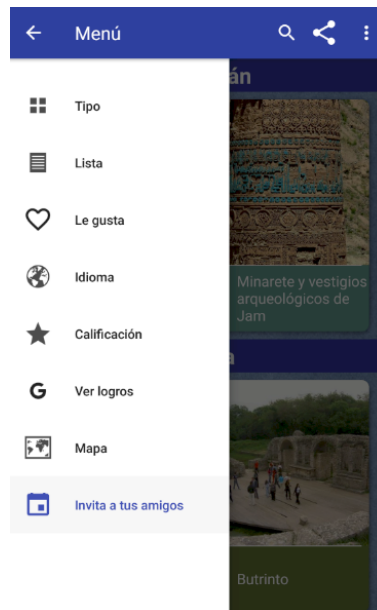


(b) Vista de un elemento

**Figura 2.5:** Interfaz de la aplicación Heritage.



- **Invita a tus amigos:** Esta opción abre una ventana desde la que es posible enviar un SMS a diferentes contactos para compartir la aplicación.



**Figura 2.6:** Menú lateral de la aplicación Worl Heritage.

Como punto destacado de la aplicación, está el mapa que permite ver las ubicaciones de los sitios, además los favoritos son otro punto a tener en cuenta ya que permiten poder visitar rápidamente los lugares preferidos.

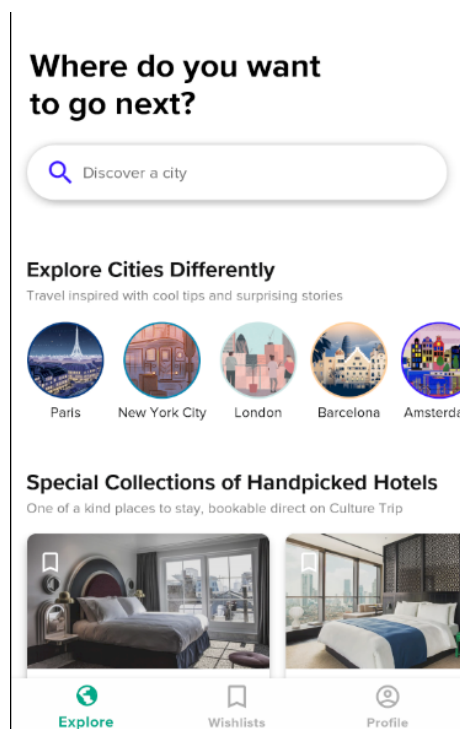
### 2.1.5. Culture Trip

Esta aplicación [5] funciona como una red social en la que las personas comparten sus experiencias sobre un lugar, estas experiencias pueden ser guías y consejos sobre sitios, como por ejemplo en donde comer, donde dormir, lugares históricos que visitar o cosas que hacer en la ciudad, sin importar si se es local o visitante.

En la aplicación al entrar, como se ve en la figura 2.7, lo primero se aprecia es un texto en grande negrita donde se pregunta **¿Donde quieres ir?**(en inglés), seguido por una barra de búsqueda que nos invita al usuario una ciudad del mundo.

Debajo se encuentra en forma de imágenes circulares ciudades famosas para comenzar a explorar a modo de sugerencias, luego más abajo algunas historias en forma de bocadillo con un pequeño botón para marcarlos y estas se presentan dinámicamente mientras se desliza la pantalla hacia abajo; la misma aplicación sugiere contenido en caso de nos saber lo que se esta buscando exactamente.

En la parte inferior hay un menú con 3 botones que está presente en todo momento en la aplicación y son:



**Figura 2.7:** Vista principal de la aplicación Culture Trip

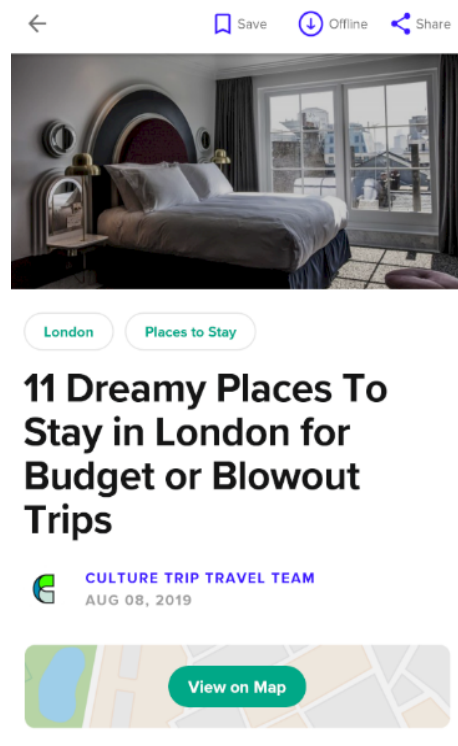
- **Explore**(Explorar): En este apartado se observan los distintos elementos que se han mencionado anteriormente, con todos los items de la aplicación en un solo lugar.
- **Wishlists**(Lista de deseos): En la lista de deseos se encuentran los elementos que se han ido guardando y que le han gustado al usuario al pulsar en el botón **me gusta**, por si quiere visitar el artículo.
- **Profile**(Perfil): En este apartado se encuentra la información de usuario, además de artículos online, ajustes de la aplicación, información de contacto, información legal y un botón de inicio de sesión.

Si se inicia sesión en la aplicación es posible editar la información de usuario; para iniciar sesión además existen con varias alternativas, como Facebook, Google o correo electrónico sin necesidad de utilizar un inicio de sesión de terceros.

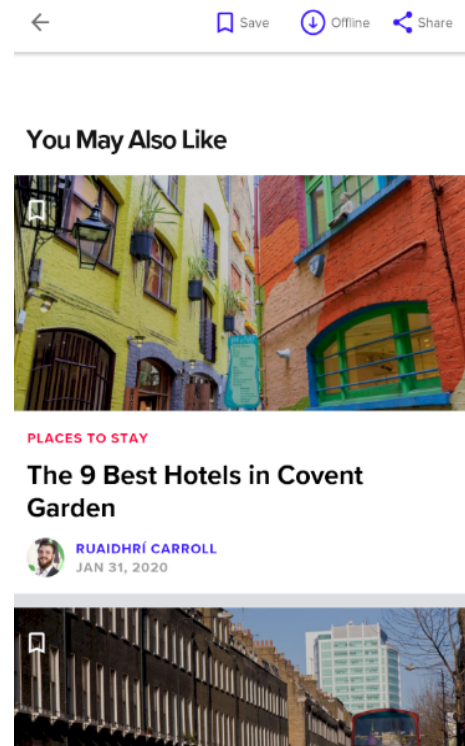
Cuando se ingresa en uno de los artículos por primera vez nos señala que es posible descargar los artículos para tenerlos sin conexión como se ve en la parte superior derecha de la figura 2.8(a), el uso de esta guía es bastante útil al entrar la primera vez, además justo en la parte superior junto al botón de descargar se tiene un botón de guardar en la lista de deseos y otro de compartir.

Los artículos tienen etiquetas para ser clasificados correctamente, como por ejemplo la ciudad o la categoría a la que pertenecen, también viene el nombre y la fecha de quién creó el artículo y un mapa donde encontrar lo que el artículo menciona (si procede), después de este mapa se encuentra todo el artículo, con un breve resumen al comienzo y luego todo el texto completo.

Después de terminar el artículo aparecen otros artículos para seguir leyendo dinámicamente en un desplazamiento vertical infinito, que son similares al que ya se ha ido leyendo y que se muestran



(a) Vista inicial de un artículo



(b) Vista de elementos relacionados

**Figura 2.8:** Interfaz de la aplicación Culture Trip.

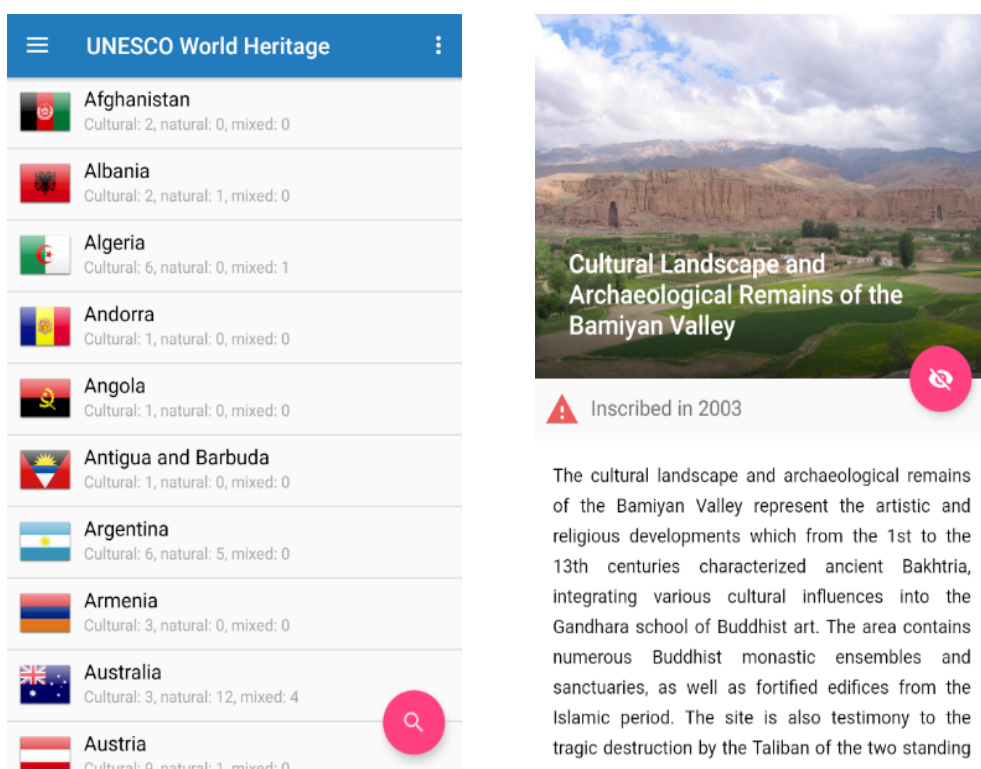
dinámicamente mientras se continua bajando.

Como algo muy destacable de esta aplicación es el hecho de presentar elementos relacionados al final del artículo para que puedan seguir navegando los usuarios en contenido que probablemente les interese.

### 2.1.6. UNESCO Heritage

Aplicación para ver patrimonios de la humanidad de cada país [6], se incluyen tanto culturales como naturales y casos en los cuales el patrimonio es tanto natural como cultural.

La aplicación posee una lista al entrar con todos los países ordenados alfabéticamente como se ve en la figura 2.9(a), para hacer más fácil la búsqueda cada uno tiene una bandera con lo que podemos hacer una búsqueda visual más rápida, debajo del nombre del país se encuentran unos detalles que indican cuantos patrimonios hay en ese país, separados en 3 categorías: “Cultural”, “Natural” y “Mixto”, además en la vista principal tiene un botón flotante en la parte inferior derecha para buscar por lugar o país.



(a) Vista inicial

(b) Vista de un elemento

**Figura 2.9:** Interfaz de la aplicación World Heritage.

En la parte superior se encuentran dos botones, uno de menú desplegable lateral y otro de 3 puntos verticales que tiene la opción de ver un patrimonio aleatorio, en el menú desplegable existe la opción

de ver cuales sitios ya se han visto, unas estadísticas de uso de la aplicación y un filtrado por regiones.

Si se elige un país para ver los patrimonios aparecerán en forma de lista igual que la vista de todos los países, una vez se entra a uno de ellos se puede ver una imagen relacionada con el patrimonio en cuestión, el nombre o título, una descripción en la parte inferior además de una imagen con la ubicación del patrimonio, por último entre la imagen del patrimonio y la descripción existe un botón para marcar el sitio como visto.

Como parte destacable de la aplicación está la opción de que elija un sitio aleatorio y que tenga las categorías de los lugares en la pantalla de inicio, además de un pequeño mapa con la ubicación que está hecho en con una imagen en vez de un mapa interactivo, esto ahorra en procesamiento del mapa y uso de una herramienta de terceros para mostrar un mapa con un solo elemento.

## 2.2. Comparativa

Se realizara una comparación de las aplicaciones anteriormente descritas y sacar de ellas los datos más importantes, el objetivo, es sacar la mejor información de cada una y lo más útil, para la aplicación a desarrollar.

### 2.2.1. Comparativa de elementos funcionales

Aplicación	Mapa	Menú	Orientación	Recomendaciones	Perfiles
ARAMEDPI	Si	Lateral	Vertical	No	No
Barcelona Art & Culture	Sí	Inferior	Vertical	No	No
Ciudades patrimonio de la humanidad	Sí	Lateral	Horizontal	No	No
Heritage	Sí	Lateral	Vertical	Aleatoria	No
Culture Trip	Sí	Inferior	Vertical	Por contexto	Sí
UNESCO Heritage	No	Lateral	Vertical	No	No

**Tabla 2.1:** Tabla comparativa de elementos funcionales de las aplicaciones vistas durante el capítulo.

Ahora vamos a proceder a analizar la tabla 2.1, en la que se pueden ver cinco categorías, conside-

radas como las más importantes teniendo en cuenta los objetivos del proyecto:

- **Mapa:** En su mayoría todas poseen un mapa, por lo que se considera una funcionalidad muy importante para este tipo de aplicaciones, con lo que es un punto casi obligatorio para cualquier aplicación similar.
- **Menú:** En este apartado la balanza se inclina más por las preferencias personales ya que algunas aplicaciones tienen un menú lateral mientras que otras tienen un menú inferior, siendo igualmente válido uno u otro.
- **Orientación:** En su mayoría la orientación es vertical por lo que se considera la más apropiada, teniendo también en cuenta el uso que hace el usuario del dispositivo, que es frecuentemente en vertical.
- **Recomendaciones:** Solo Culture Trip y Heritage poseen esta característica, aunque, de manera diferente, en el primer caso tenemos recomendaciones basadas en lo que estamos viendo en ese momento, sin embargo, en el segundo caso tenemos recomendaciones aleatorias sin un algoritmo de recomendación detrás.
- **Perfiles:** En su mayoría las aplicaciones mencionadas no necesitan tener información del usuario para poder funcionar, con lo que no necesitan hacer uso de un perfil, en el caso de la que sí lo tiene que es Culture Trip, podemos ver que hace uso de esto para guardar las preferencias del usuario y presentar funcionalidades similares a una red social.

## 2.2.2. Comparativa de elementos audiovisuales

Aplicación	Audioguía	Multilenguaje	Imágenes	Ambito
ARAMEDPI	No	Sí	Sí	España
Barcelona Art & Culture	No	No	Sí	Barcelona
Ciudades patrimonio de la humanidad	Sí	No	Sí	España
Heritage	No	Sí	Sí	Países
Culture Trip	No	No	Sí	Ciudades
UNESCO Heritage	No	No	Sí	Países

**Tabla 2.2:** Tabla comparativa de elementos audiovisuales entre las distintas aplicaciones con fines similares a la de este trabajo.

Tras un análisis de las características más relevantes de la aplicación en los aspectos audiovisuales en la tabla 2.2, se determina que son los siguientes que pasaremos a analizar:

- **Audioguía:** No es un elemento muy extendido, aunque sí interesante, pero que no se tendrá en cuenta para el alcance de este TFG.

- **Multilinguaje:** Es una característica que como vemos no está demasiado extendida, pero que teniendo en cuenta que llevando la aplicación a más idiomas el público objetivo también será mayor, se decide tener en cuenta el multilinguaje como una característica importante para llegar al máximo público posible.
- **Imágenes:** Las imágenes sí que es uno de los puntos más importantes, ya que todas las lo tienen, por lo cual, este punto es imprescindible entre aplicaciones de este tipo y debe tenerse en cuenta para este proyecto.





# ANÁLISIS

---

Una vez realizado el estudio sobre aplicaciones similares a la propuesta en este tfg, se define el alcance de este proyecto precisando, tanto los requisitos funcionales como no funcionales que debe proporcionar la aplicación.

## 3.1. Requisitos funcionales

- RF-1.**— La aplicación debe permitir registrar usuarios.
- RF-2.**— La aplicación solo debe permitir acceder a usuarios registrados.
- RF-3.**— El usuario debe poseer una área privada para poder cambiar sus datos y ajustes.
- RF-4.**— La aplicación debe tener una pantalla principal donde se dé acceso a los tipos de dato: histórico, artístico, cultural, natural y gastronómico.
- RF-5.**— Ofrecer una opción de búsqueda por filtros: por nombre, por distancia, por autor, por localización.
- RF-6.**— Pantalla donde se muestre la información (meta-datos) asociada a un objeto o entidad.
- RF-7.**— Recomendación de un sitio en la pantalla principal, para que el usuario descubra un elemento nuevo cada vez que utiliza la aplicación.
- RF-8.**— Pantalla de generación de una ruta turística personalizada.
- RF-9.**— Pantalla con mapa con la ubicación de los POI.
- RF-10.**— Desde cada objeto se debe poder acceder a la ubicación.
- RF-11.**— Los objetos deben tener al menos una imagen cada uno.
- RF-12.**— La aplicación debe permitir guardar los elementos favoritos a través de un botón en la pantalla de información del objeto.
- RF-13.**— La aplicación debe guardar los favoritos fuera de la aplicación y asociado al usuario autenticado.
- RF-14.**— La aplicación debe tener un menú inferior en el que poder navegar entre las opciones, (pantalla inicial, mapa, favoritos, configuración).
- RF-15.**— La aplicación debe tener un botón de cambio de idioma en la configuración.
- RF-16.**— Cada objeto debe ofrecer como mínimo el nombre, la descripción, y una imagen para mostrar en la pantalla inicial para poder ser diferenciados de los otros.
- RF-17.**— La aplicación no podrá orientarse en horizontal.
- RF-18.**— Los mapas deben ser interactivos y no en imágenes.
- RF-19.**— Si existe, debe haber un enlace al sitio web correspondiente.

## 3.2. Requisitos no funcionales

**RNF-1.**— La base de datos de la aplicación se invocarán a través de una API REST en un servidor externo.

**RNF-2.**— Los textos de la aplicación deben estar disponibles tanto en español como en inglés.

**RNF-3.**— La aplicación debe estar disponible para Android a partir de la version 6 (Marshmallow).

**RNF-4.**— La aplicación debe seguir patrones de diseño de Google (Material Design).

**RNF-5.**— La lógica compleja se hará en un servidor externo, minimizando el impacto de rendimiento en la aplicación.

**RNF-6.**— Se utilizará caché para la API y minimizar así su tiempo de respuesta.

## 3.3. Casos de uso

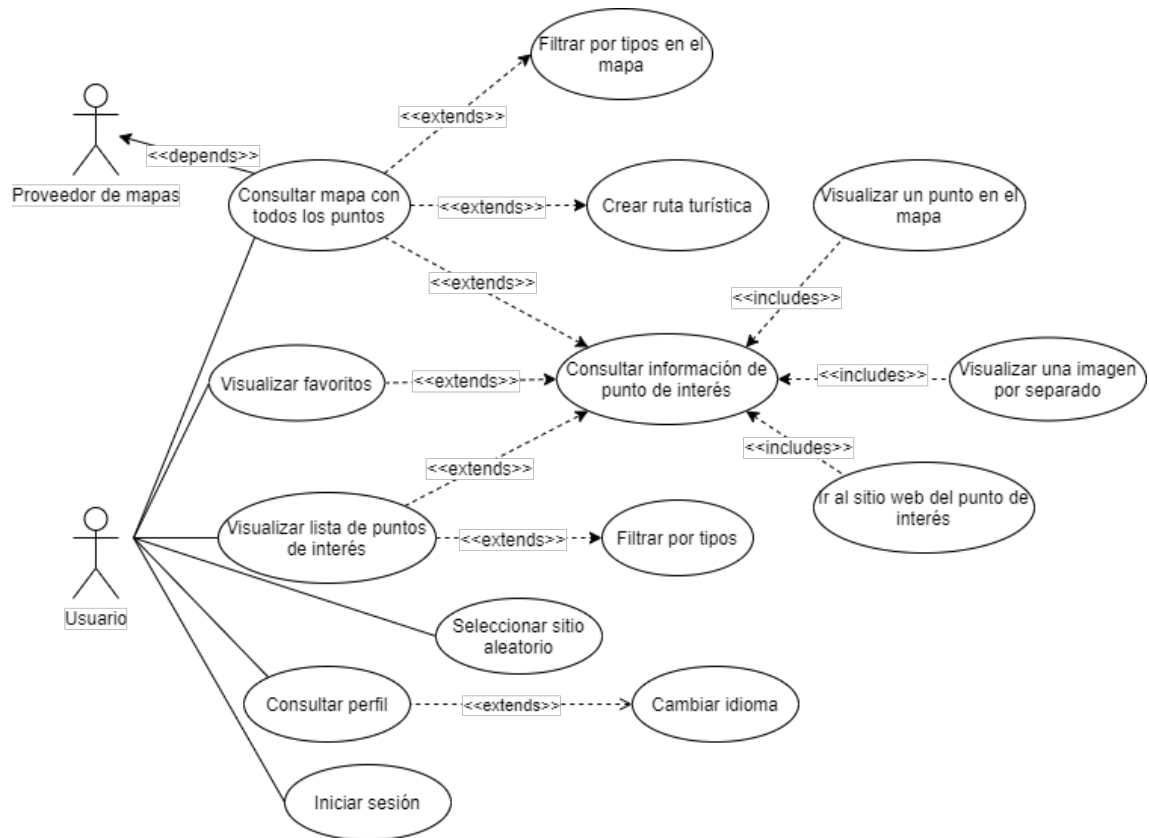
### 3.3.1. Actores en el sistema

Se denominan actores a los personajes o entidades que participan en un diagrama de casos de uso. En este proyecto aparecen los siguientes:

- **Usuario:** persona que usa la aplicación.
- **Proveedor de mapas:** servicio web para obtener mapas para colocar los POI.

### 3.3.2. Diagrama de casos de uso

El diagrama de casos de uso que se muestra a continuación nos indica las acciones que el usuario puede realizar dentro de la aplicación, además de como dependen uno de otro.



**Figura 3.1:** Diagrama de casos de uso

### 3.3.3. Descripción de casos de uso

En este punto se pasa a definir cada uno de los casos de uso de forma más detallada y los distintos escenarios posibles para cada acción.

<b>Identificador del caso de uso: CU-1</b>
<b>Nombre del caso de uso:</b> Registrarse
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> No estar dentro de una sesión.
<b>Garantía de éxito, (Postcondiciones):</b> Se crea el usuario y se puede utilizar para iniciar sesión.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. El usuario escoge la opción de <b>Registrarse</b>.</li> <li>3. El usuario introduce todos los datos solicitados por el formulario de registro.</li> <li>4. El usuario envía los datos y se completa el registro, el usuario ya puede utilizarse para guardar datos.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>4a. El dispositivo se encuentra sin conexión a internet y los datos no pueden enviarse.</li> </ol>

**Tabla 3.1:** Caso de uso de registro del usuario

<b>Identificador del caso de uso: CU-2</b>
<b>Nombre del caso de uso:</b> Iniciar sesión
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> No estar dentro de una sesión.
<b>Garantía de éxito, (Postcondiciones):</b> Se inicia sesión con el usuario y su información personal es guardada.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. El usuario escoge la opción de <b>Iniciar sesión</b>.</li> <li>3. El usuario introduce su nombre de usuario y contraseña.</li> <li>4. El sistema autentica si el inicio de sesión ha sido correcto.</li> <li>5. El sistema permite al usuario el acceso al usuario al resto de la aplicación.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>4a. El dispositivo se encuentra sin conexión a internet y no puede autenticarse.</li> <li>4b. El usuario introduce los datos incorrectamente y no puede autenticarse.</li> </ol>

Tabla 3.2: Caso de uso de inicio de sesión

<b>Identificador del caso de uso: CU-3</b>
<b>Nombre del caso de uso:</b> Visualizar lista de POIs
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> Haber iniciado sesión en la aplicación.
<b>Garantía de éxito, (Postcondiciones):</b> Se visualiza la lista de POIs.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. El sistema carga desde un servicio externo los datos correspondientes.</li> <li>3. El sistema carga los sitio en la pantalla principal.</li> <li>4. El usuario visualiza los lugares con sus imágenes.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>2a. El dispositivo se encuentra sin conexión a internet y el sistema no puede acceder al servicio.</li> <li>3a. No se cargan los POIs.</li> <li>4a. El usuario no visualiza la lista.</li> </ol>

Tabla 3.3: Caso de uso de visualización lista de POIs

<b>Identificador del caso de uso: CU-4</b>
<b>Nombre del caso de uso:</b> Visualizar favoritos
<b>Actores involucrados(Actores primarios):</b> Usuario.
<b>Precondiciones:</b> El usuario debe haber pulsado en la sección de favoritos del menú y debe haber iniciado sesión
<b>Garantía de éxito, (Postcondiciones):</b> Aparece la lista de favoritos
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario pulsa en el icono de favoritos.</li> <li>2. El sistema carga de la memoria local los id que el usuario ha guardado como favoritos.</li> <li>3. El sistema solicita al servicio externo los objetos.</li> <li>4. El usuario visualiza la lista de los POIs favoritos.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>2a. El dispositivo se encuentra sin conexión a internet y el sistema no puede acceder al servicio.</li> <li>3a. No se cargan los POIs favoritos.</li> <li>4a. El usuario no visualiza la lista con sus favoritos.</li> </ol>

**Tabla 3.4:** Caso de uso de visualización de favoritos

<b>Identificador del caso de uso: CU-5</b>
<b>Nombre del caso de uso:</b> Cambiar idioma
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> El usuario debe haber iniciado sesión.
<b>Garantía de éxito, (Postcondiciones):</b> Todos los textos de la aplicación se cambian al nuevo idioma seleccionado.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la sección de ajustes.</li> <li>2. El usuario selecciona cambio de idioma.</li> <li>3. El usuario selecciona el nuevo idioma para los textos.</li> <li>4. Todos los textos se actualizan.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>3a. El usuario vuelve atrás sin seleccionar ningún idioma.</li> <li>4a. El sistema no debe cambiar los textos en este flujo.</li> </ol>

**Tabla 3.5:** Caso de uso cambio de idioma

<b>Identificador del caso de uso: CU-6</b>
<b>Nombre del caso de uso:</b> Seleccionar sitio aleatorio
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> El usuario se tiene que encontrar en la vista inicial y haber iniciado sesión para acceder al sistema.
<b>Garantía de éxito, (Postcondiciones):</b> Se abre un POI aleatorio con toda su información completa.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario se encuentra en el menú inicial o pulsa el botón para volver a este.</li> <li>2. El usuario pulsa el botón de selección aleatoria.</li> <li>3. El sistema carga el id aleatorio con toda su información.</li> <li>4. Se abre un POI aleatorio.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <b>3a.</b> No se tiene conexión a internet y el sistema no puede abrir el POI.

Tabla 3.6: Caso de uso de selección sitio aleatorio

<b>Identificador del caso de uso: CU-7</b>
<b>Nombre del caso de uso:</b> Consultar mapa con todos los puntos
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> El usuario debe estar dentro de la aplicación, tener conexión a internet y haber iniciado sesión.
<b>Garantía de éxito, (Postcondiciones):</b> Se visualizan todos los puntos en el mapa de la aplicación.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la sección del mapa.</li> <li>2. La aplicación tendrá una posición inicial para el mapa en la que se vean todos los puntos al mismo tiempo y la enviará al servicio de mapas.</li> <li>3. El servicio de mapas devolverá el mapa.</li> <li>4. Las localizaciones se solicitan a la base de datos externa y solo las visibles en el mapa actual.</li> <li>5. El sistema pinta los POIs en el mapa.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <b>3a.</b> No hay conexión a internet y el mapa no se puede solicitar. <b>4b.</b> La base de datos externa no funciona y el sistema no puede dibujar sobre el mapa los POIs.

Tabla 3.7: Caso de uso de consulta del mapa con todos los puntos

<b>Identificador del caso de uso: CU-8</b>
<b>Nombre del caso de uso:</b> Consultar información de un POI.
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> El usuario debe estar dentro de la vista principal o en favoritos y seleccionar un POI, además de tener internet para consultar los POIs y haber iniciado sesión.
<b>Garantía de éxito, (Postcondiciones):</b> El POI muestra toda la información además de sus imágenes.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la lista de POIs.</li> <li>2. El usuario pulsa selecciona un POI</li> <li>3. El sistema recoge el identificador del punto y consulta en base de datos la información del punto para presentarla al usuario.</li> <li>4. Se pinta en pantalla la información al usuario del POI.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>1a. El usuario no puede acceder a la lista de puntos porque está sin conexión a internet.</li> <li>3b. <b>No se puede conectar con la base de datos y recuperar la información del POI.</b></li> <li>4b. <b>Se presenta un mensaje de error al usuario alertando de que ha fallado la conexión.</b></li> </ol>

Tabla 3.8: Caso de uso de consulta de información de un POI

<b>Identificador del caso de uso: CU-9</b>
<b>Nombre del caso de uso:</b> Filtrar por tipos en la lista.
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> El usuario debe estar conectado a internet, debe estar en la vista principal y haber iniciado sesión en la aplicación.
<b>Garantía de éxito, (Postcondiciones):</b> El usuario solo ve los POIs que se han filtrado.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario entra en la vista principal.</li> <li>2. El sistema busca las categorías en la base de datos.</li> <li>3. El usuario pulsa en una categoría.</li> <li>4. El sistema busca los POIs que pertenecen a esa categoría.</li> <li>5. El sistema pinta los POIs filtrados.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>2a. No hay internet y el sistema no puede acceder a la base de datos.</li> <li>3a. Se presenta un mensaje de error al usuario de fallo en la conexión.</li> <li>3b. El usuario pulsa atrás y sale de la aplicación.</li> <li>4c. Falla la conexión y el usuario no puede ver los POIs de la categoría.</li> </ol>

Tabla 3.9: Caso de uso de filtro por tipos en lista



<b>Identificador del caso de uso: CU-10</b>
<b>Nombre del caso de uso:</b> Filtrar por tipos en el mapa.
<b>Actores involucrados(Actores primarios):</b> Usuario y proveedor de mapas.
<b>Precondiciones:</b> El usuario debe disponer de internet y haber iniciado sesión.
<b>Garantía de éxito, (Postcondiciones):</b> Se visualiza el mapa con los POIs en él.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la sección de mapa.</li> <li>2. El sistema solicita al proveedor de mapas por el mapa.</li> <li>3. El sistema solicita los POIs que caben en la vista inicial del mapa.</li> <li>4. El sistema pinta los POIs.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>2a. El usuario no tiene conexión a internet y el mapa no puede cargarse.</li> <li>3b. El usuario no puede conectar con la base de datos y los puntos no se cargan.</li> </ol>

Tabla 3.10: Caso de uso de filtrar por tipos en el mapa

<b>Identificador del caso de uso: CU-11</b>
<b>Nombre del caso de uso:</b> Visualizar un punto en el mapa.
<b>Actores involucrados(Actores primarios):</b> Usuario y proveedor de mapas
<b>Precondiciones:</b> El usuario debe disponer de conexión a internet, encontrarse dentro de la información de un POI y haber iniciado sesión.
<b>Garantía de éxito, (Postcondiciones):</b> El usuario visualiza el mapa
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario está visualizando un POI.</li> <li>2. El usuario pulsa el botón para visualizar el POI en el mapa.</li> <li>3. Se solicita el mapa al proveedor de mapas para pintar el punto.</li> <li>4. Se utiliza la información ya disponible sobre el POI para pintarlo en el mapa.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>3a. Falla la conexión con el proveedor de mapas.</li> <li>4a. Aparece un mensaje de error de conexión.</li> </ol>

Tabla 3.11: Caso de uso de visualización de un punto en el mapa

<b>Identificador del caso de uso: CU-12</b>
<b>Nombre del caso de uso:</b> Visualizar imágenes por separado
<b>Actores involucrados(Actores primarios):</b> Usuario.
<b>Precondiciones:</b> El usuario debe estar visualizando un POI y haber iniciado sesión en la aplicación.
<b>Garantía de éxito, (Postcondiciones):</b> El usuario visualiza las imágenes en carrete sin información adicional.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario se encuentra dentro de la información de un POI.</li> <li>2. El usuario pulsa sobre la imagen.</li> <li>3. La imagen se agranda y se ocultan todos los demás datos del POI.</li> </ol>
<b>Extensiones(Flujos alternativos):</b>

**Tabla 3.12:** Caso de uso de visualización de imágenes por separado

<b>Identificador del caso de uso: CU-13</b>
<b>Nombre del caso de uso:</b> Ir al sitio web del POI
<b>Actores involucrados(Actores primarios):</b> Usuario
<b>Precondiciones:</b> El usuario debe estar dentro de la información del POI y haber iniciado sesión en la aplicación.
<b>Garantía de éxito, (Postcondiciones):</b> Se abre el navegador y se visualiza el sitio web relacionado al POI.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"> <li>1. El usuario se encuentra dentro de la información de un POI.</li> <li>2. El usuario pulsa el botón correspondiente para visitar el sitio web.</li> <li>3. Se abre el navegador con la página web solicitada.</li> </ol>
<b>Extensiones(Flujos alternativos):</b> <ol style="list-style-type: none"> <li>3a. Falla la conexión y la página web no puede cargarse</li> </ol>

**Tabla 3.13:** Caso de uso de redirección al sitio web del POI

<b>Identificador del caso de uso: CU-14</b>
<b>Nombre del caso de uso:</b> Crear ruta turística
<b>Actores involucrados(Actores primarios):</b> Usuario y proveedor de mapas
<b>Precondiciones:</b> El usuario debe estar dentro de la vista de mapa y haber iniciado sesión.
<b>Escenario principal de éxito:</b> <ol style="list-style-type: none"><li>1. El usuario debe estar en la vista de mapa.</li><li>2. Debe pulsar sobre el botón de crear ruta.</li><li>3. El usuario selecciona los POIs que le desea visitar.</li><li>4. El mapa muestra las conexiones óptimas entre los POIs.</li></ol>
<b>Extensiones(Flujos alternativos):</b>

Tabla 3.14: Caso de uso de crear ruta turística



# DISEÑO

---

Se plantean los diseños a seguir en el desarrollo, la lógica del servidor que provee a la aplicación, los esquemas relacionados con la base de datos y las clases que contendrá la aplicación, así como un diseño de la arquitectura.

## 4.1. Prototipo de la aplicación

En el diseño de la aplicación se deciden agregar algunas funcionalidades interesantes vistas en las aplicaciones analizadas en el capítulo 2:

De la aplicación ARAMEDPI vista en el apartado 2.1.1 se extrae la funcionalidad de multilenguaje, la cual bastante útil para llegar al público extranjero que pueda visitar la Comunidad de Madrid.

### 4.1.1. Logotipo de la aplicación

Teniendo en cuenta la necesidad de relacionar la Comunidad de Madrid con Latinoamérica se ha planteado un logotipo utilizando como base el mapa de Latinoamérica y la bandera de la Comunidad de Madrid, conectando así estas dos regiones en una sola imagen.



(a) Logotipo cuadrado con las esquinas redondeadas



(b) Logotipo circular

**Figura 4.1:** Logotipos de la aplicación.

### 4.1.2. Prototipo de la interfaz

El diseño de la interfaz de usuario de la aplicación debe seguir un patrón de diseño que sea conocido por el usuario, a fin de facilitar el uso del mismo, por esta razón se escoge el patrón de diseño Material Design creado por la empresa Google, el cual es utilizado por muchas aplicaciones y que resulta familiar para el usuario, se pueden ver las capturas en el capítulo 7 donde se encuentran los Anexos.

## 4.2. Arquitectura de la aplicación

El sistema se desarrollará utilizando el SDK de Android, con compatibilidad a partir de la API nivel 23, o lo que es lo mismo, a partir de la versión 6.0 Marshmallow. La arquitectura para el frontend seguirá el patrón de diseño MVC, este es uno de los patrones más comunes en el desarrollo de aplicaciones Android y permite una clara navegación en el código. También se añade una parte backend con la lógica de negocio dividida en varios servidores utilizando Spring para crear una arquitectura de microservicios [7]. El diagrama completo en el que vemos la arquitectura y como se comunica la aplicación con los microservicios se puede ver en la figura 4.2.

El sistema estará dividido entonces en dos partes, de manera que la información y la representación de la información quede separada:

- **Backend:** La parte backend posee una arquitectura de microservicios en la que se encuentran varios servidores fundamentales para el necesarios para el funcionamiento de la aplicación y otros complementarios que añaden funcionalidades extras.

El backend está pensado para ser totalmente independiente de como se construya el frontend, ya sea una página web, una aplicación Android o una aplicación en iOS.

El backend utiliza el framework de Spring [8] junto a las herramientas de Spring Boot [9] y Spring Cloud [10] para el desarrollo de arquitecturas de microservicios [7], esta arquitectura permite elaborar distintos servidores que serán independientes y que pueden ir añadiendo nuevas funcionalidades sin tener que detener todo el sistema o en caso de fallo permitir que no todo el sistema se caiga, solo caerá la parte que ha fallado.

La seguridad del backend es a través de OAuth2 [11] haciendo que este tenga uno de los estándares más usados y seguros hasta la fecha [12].

Estarán disponibles dos servicios, uno en el que se realizará la consulta de datos a base de datos para las localizaciones y otro que será el encargado de generar las rutas personalizadas para cada usuario, esta acción es un poco más pesada y de esta manera no influye en el rendimiento del otro servidor.

Además estará el servidor de autenticación y que a la vez servirá para consultar la información relacionada con los usuarios.

Por último dos servidores adicionales, el primero es un servidor de enrutado de las peticiones que redirige las peticiones y hace que para la aplicación desde haya virtualmente solo un servidor [13]; el otro es un servidor de descubrimiento, este nos permite ver los diferentes servidores que están activos [14], así como facilitar la comunicación interna entre ellos.

- **Frontend:** El frontend solo existe temporalmente para Android, desarrollado de forma nativa siguiendo el patrón de diseño MVC [15] y las directrices de Google en cuanto diseño llamadas Material Design [16].

## 4.3. Modelo de datos

Los modelos de datos que se utilizarán dentro de la aplicación serán los que se pueden ver en la figura 4.3, en el que se ha puesto en el centro el modelo POI que es el centro de la aplicación, a su alrededor tenemos modelos que o bien se utilizan dentro de este o lo incluyen de algún modo, vamos a ver cada uno de ellos por separado:

- **POI:** El objeto principal de la aplicación, este representa los datos los puntos de interés antes de que sea vista toda la información, o sea esta es la información que se va a utilizar para los mapas y las listas.
- **DetailedPOI:** Es la extensión del POI, con los datos más específicos como las imágenes o el sitio web asociado, si dispone de este.
- **Location:** La localización del punto para representarlo en el mapa, este es el objeto más importante para poder representar los puntos en el mapa.
- **Category:** La categoría a la que pertenece el punto, esto es importante para poder dividir los puntos en clases diferentes y poder filtrar en el caso del mapa.
- **Route:** La ruta, puede contener varios puntos y es generada en tiempo de ejecución según las preferencias del usuario, ésta no posee ningún tipo de identificador único ya que no se guarda, por lo que no es necesario diferenciarla de otras rutas.
- **User:** El usuario, este es uno de los pilares de la aplicación y en el que se guardan sus favoritos, su imagen de perfil, su nombre y el idioma de preferencia.

## 4.4. Diagramas entidad relación y base de datos

El diagrama entidad-relación que se ve en la figura 4.4, se diferencia un poco de los modelos ya que algunos de estos no se guardan en base de datos, por lo que vamos a hacer un análisis de las diferencias que hay en este diagrama:

- **Language:** Representa un idioma y así poder escoger en la aplicación cual es el idioma que deseamos.
- **Translation:** Representa un texto en alguno de los idiomas disponibles, esta traducción tiene un código asociado para saber cual es el texto que representa, este código no es único ya que los mismos textos traducidos en otros idiomas tendrán el mismo pero con otro idioma.
- **POI:** El punto de interés en este caso es una sola entidad única y que posee en su interior todos los datos, no hay una versión reducida sino que esa versión vista en 4.3 se genera a partir de una consulta de una de solo algunos de los campos.
- **User:** En este caso el usuario posee la contraseña y la fecha de creación como datos extras, es necesario que la contraseña esté para poder autenticarse.

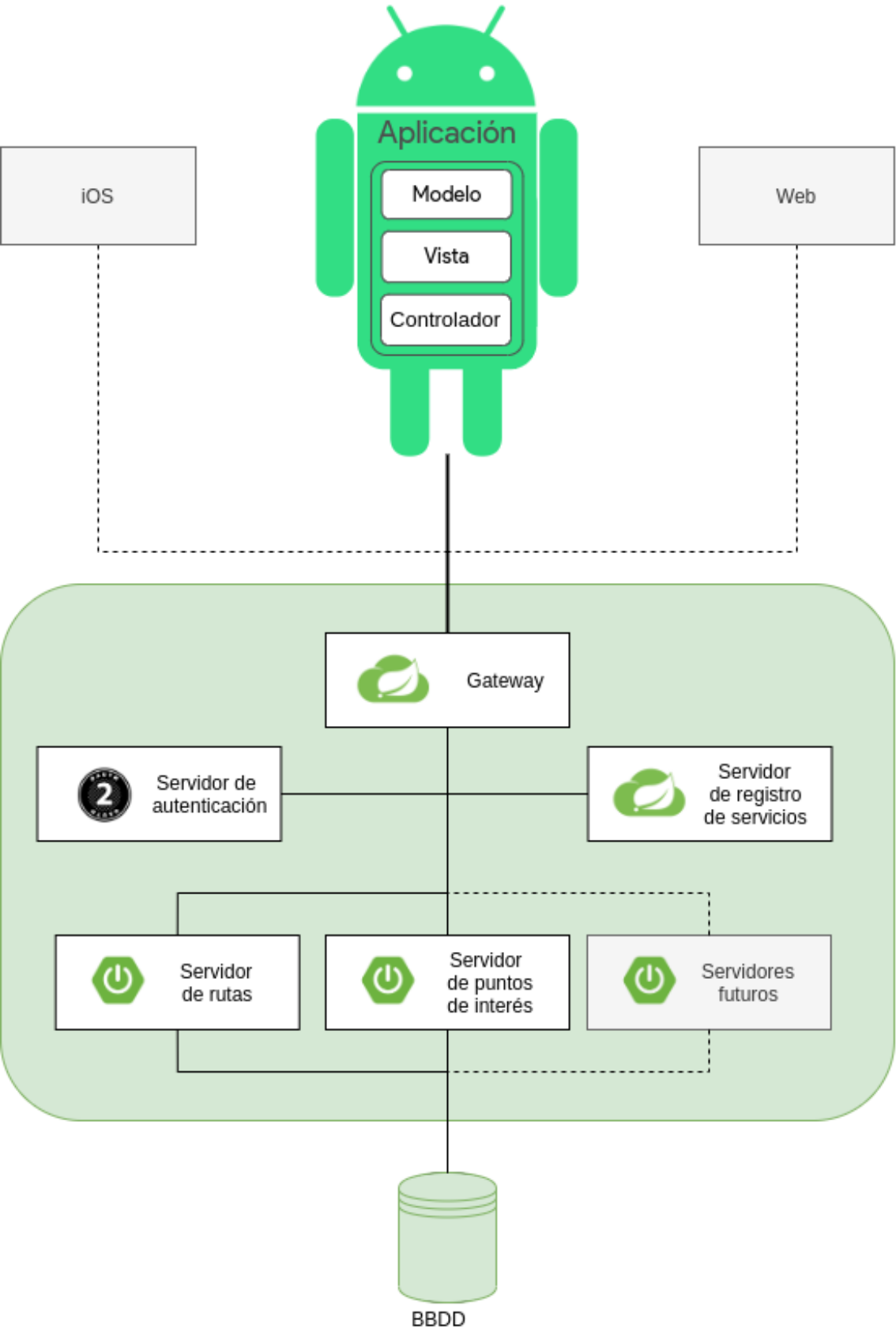


Figura 4.2: Arquitectura del sistema



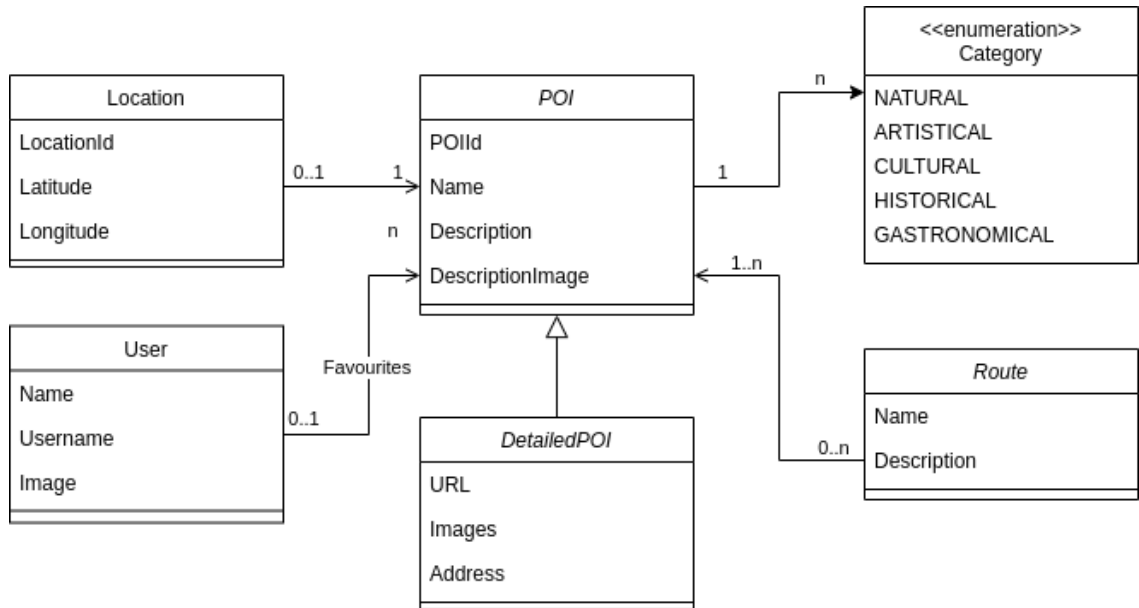


Figura 4.3: Modelos de datos.

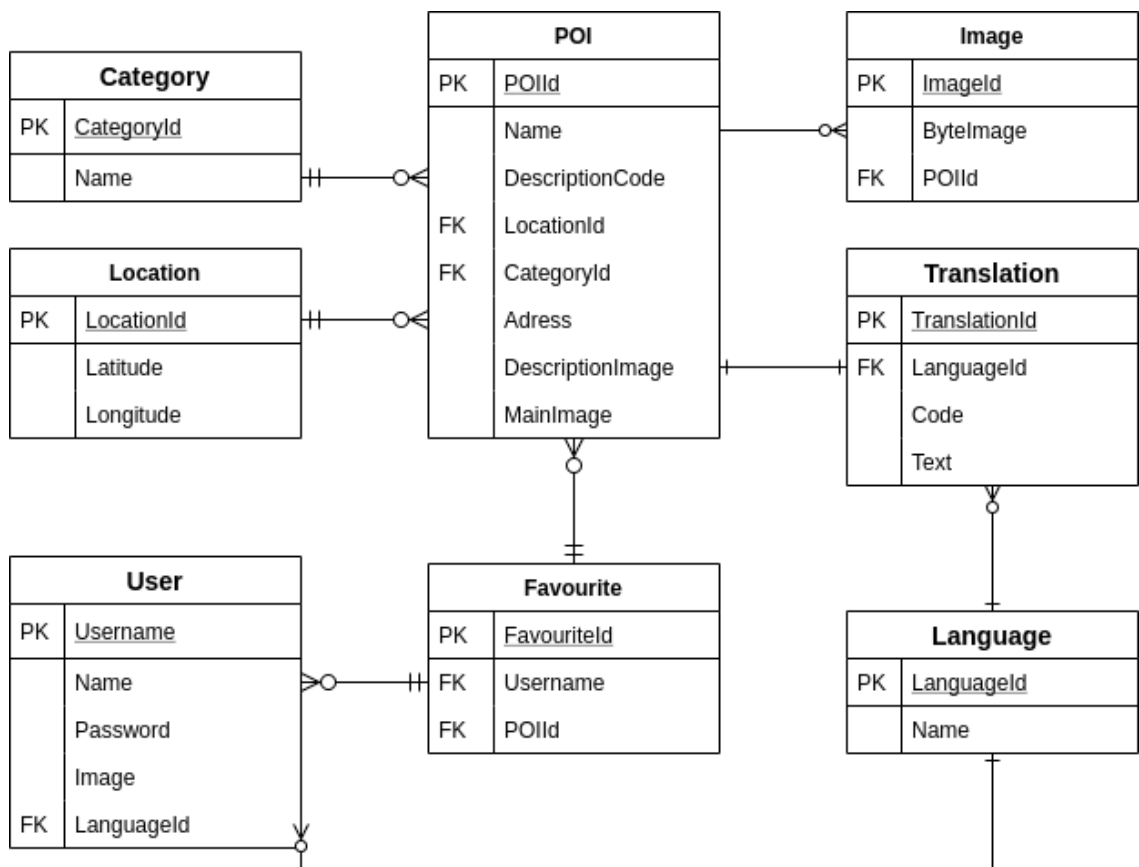


Figura 4.4: Diagrama entidad relación.



## DESARROLLO

---

### 5.1. Herramientas y entornos

Los entornos de desarrollo son importantes para poder crear de manera mucho más sencilla y rápida las aplicaciones que se necesitan, en este caso se usarán las siguientes herramientas:

- **Android Studio [17]:** El entorno de desarrollo por defecto para aplicaciones android nativas será donde se lleve a cabo la aplicación y donde se diseñen las pantallas y todo lo relacionado con la interfaz del usuario.
- **Spring Tool Suite [18]:** Esta herramienta está basada en el entorno de desarrollo Eclipse, el cual será utilizado para la creación de los microservicios de la lógica de negocio más pesada.
- **PgAdmin [19]:** Para administrar la base de datos se utilizará postgresql [20], la herramienta PgAdmin proporciona una interfaz cómoda para manejar las operaciones de la base de datos que sean necesarias.

### 5.2. Librerías

Dentro de cada entorno de desarrollo existen algunas librerías que serán utilizadas para simplificar el código y ser más eficientes a la hora de crear nuestra aplicación:

- **Spring Boot [9]:** Esta es una librería que permite tener algunas herramientas como un servidor tomcat embebido.
- **Spring Cloud [10]:** Posee algunas herramientas útiles para la comunicación entre microservicios, como el servicio de descubrimiento de servidores activos, el enrutado de las peticiones, control de errores entre microservicios entre otras herramientas que no serán utilizadas en la aplicación.
- **Android SDK [21]:** La librería más básica para programar en Android y proporcionada por Google, su programación está orientada a eventos.
- **Java [22]:** Todas las herramientas mencionadas anteriormente están hechas para el lenguaje de programación Java, aunque existen variantes utilizando otros lenguajes este es el más común en este caso.

### 5.3. Microservidores

Para crear un servidor en Spring Boot es bastante sencillo, se puede utilizar la ayuda de la herramienta online Spring Initializr [23], esta herramienta ya propone un proyecto con lo mínimo para comenzar y desplegar.

La clase main de la aplicación es donde se ponen las anotaciones que dirán como estará configurada o donde está la configuración de la aplicación, como se ve en la figura 5.1 esta la anotación principal que sería `@SpringBootApplication` [24], mientras que la anotación `@EnableEurekaClient` [25] sirve para habilitar la conexión con el servicio de descubrimiento y la anotación `@EnableCaching` [26] para habilitar la caché de spring y que algunas llamadas, con la anotación apropiada, respondan de manera más rápida gracias a esta caché.

En el caso de que se quiera además configurar donde debe buscar Spring los paquetes que contienen los componentes, las entidades o los repositorios se puede indicar como se ve en la figura 5.2.

```
@SpringBootApplication
@EnableEurekaClient
@EnableCaching
public class LocationServiceApp {
    public static void main(String[] args) {
        SpringApplication.run(LocationServiceApp.class, args);
    }
}
```

**Figura 5.1:** Clase principal de la aplicación con Spring Boot

```
@SpringBootApplication
@EnableEurekaClient
@EnableCaching
@ComponentScan(basePackages = {"com.uam.es.madlat"})
@EntityScan( basePackages = {"com.uam.es.madlat"} )
@EnableJpaRepositories("com.uam.es.madlat")
public class LocationServiceApp {
    public static void main(String[] args) {
        SpringApplication.run(LocationServiceApp.class, args);
    }
}
```

**Figura 5.2:** Clase principal de la aplicación con Spring Boot con configuración específica.

### 5.3.1. Acceso a base de datos

Los datos se acceden desde la aplicación a través de unas llamadas a APIs REST 5.3.2 que van a los microservicios, los cuales serán los encargados de gestionar cada uno el acceso a la base de datos, se utilizará la librería Spring Data JPA [27] para ello, con esto se logra crear un repositorio con el que se accede a los datos simplemente creando una interfaz, además a través de los nombres de los métodos se pueden crear consultas sin escribir código SQL [28], se observa en la figura 5.3 un ejemplo de estas interfaces.

```
@Repository
public interface POIRepository extends JpaRepository<POI, Long>{

    List<POI> findByCategory(Category category);

    List<POIBasicBean> findByLocationNotNull();

    List<POIBasicBean> findByCategoryAndLocationNotNull(Category category);
}
```

Figura 5.3: Ejemplo de uso de la librería Spring JPA.

Para la creación de las tablas se utiliza la librería javax.persistence [29] que junto a las librerías de Spring crean automáticamente las tablas directamente en la base de datos sin tener que crearse estas manualmente como se ve en la figura 5.4.

La conexión a la base de datos también es bastante sencilla y se realiza en apenas unos segundos, para ello se ponen los datos de conexión en el fichero application.yml del proyecto Spring Boot y este se encargará por si solo de crear todo lo necesario para conectarse con esa base de datos, ya que además se indicara que clase de driver debe usar pues este varía dependiendo del tipo de base de datos, en la figura 5.5 se observa esta configuración, además también se puede ver otro tipo de configuración como el puerto en el que se va a desplegar el servidor y la conexión con el servicio de descubrimiento.

### 5.3.2. Creación de API REST

Para crear la API REST se utiliza la anotación de Spring que es @RestController [30], con esto la clase se convierte en un Bean de Spring [31] con algunos añadidos como el poder aceptar peticiones HTTP [32], en la figura 5.6 vemos como se crea el controlador y se reciben todas las peticiones que vengan con **/poi**, además en la misma figura 5.6 podemos ver como más abajo hace otro mapeo, pero en este caso la url sería **/poi/category/{category}**, donde {category} es el nombre de un valor del enumerado en la clase **Category**, la variable se recoge con la anotación @PathVariable.

```
@Entity
@Table(name = "POI")
@Data
public class POI {

    @Id
    @GeneratedValue
    private Long poiId;

    @Column(nullable = false)
    private String name;

    @Column(nullable = false)
    private String region;

    @Lob
    @Column(nullable = false)
    private String description;

    @OneToOne(optional = false, cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    private Image image;

    @ManyToOne(optional = false, cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    private Location location;

    @Enumerated(EnumType.ORDINAL)
    private Category category;

    @Column
    private String url;

    @OneToOne(optional = false, cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    private Image miniature;

    @Column
    private Integer timeToSpend;
}
```

Figura 5.4: Creación de tablas de manera automática con el uso de etiquetas.

```

server:
  port: 8110
logging:
  level:
    jdbc:
      resultsettable: DEBUG
      sqltiming: DEBUG
    com.uam: ERROR
spring:
  application:
    name: users-service
  main:
    banner-mode: 'OFF'
  thymeleaf:
    cache: false
#drop n create table again, good for testing, comment this in production
jpa:
  hibernate:
    ddl-auto: update
    show-sql: true
    properties.hibernate.dialect : org.hibernate.dialect.PostgreSQLDialect

# default connection pool
datasource:
  initialization-mode: always
  hikari:
    connectionTimeout: 20000
    maximumPoolSize: 5

## PostgreSQL
url: jdbc:postgresql://localhost:5432/madlat
username: postgres
password: ■■

eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/

```

**Figura 5.5:** Archivo de configuración de application.yml de Spring Boot.

```

@RestController
@RequestMapping("/poi")
public class POIController {

    @Autowired
    POIService poiService;

    @Cacheable(value = "poi-elements-by-category")
    @GetMapping("/category/{category}")
    public List<POIBean> getPOIByCategory(@PathVariable("category") Category category) {
        return poiService.getPOIByCategory(category);
    }
}

```

**Figura 5.6:** Ejemplo de una API REST con Spring Framework

Para crear un endpoint en particular se utilizan las notaciones `@GetMapping`, `@PutMapping`, `@PostMapping`, `@DeleteMapping`, para crear los metodos HTTP respectivos, como se ve en la figura 5.7 además se asignar el tipo de datos que consume y produce, pero esto último es opcional.

```
@PutMapping(path =("/{poiId}", consumes = "application/json", produces = "application/json")
public POIBean upadte(@PathVariable("id") Long poiId, @RequestBody POIBean poi) {
    return poiService.savePOI(poi);
}
```

**Figura 5.7:** Ejemplo de un endpoint para la API REST

## 5.4. Android

El desarrollo en Android es mucho más sencillo y se sigue el patrón de diseño de modelo-vista-controlador(MVC), haciendo un uso además, de la guía que ofrece Google para desarrollar con su SDK [21].

Se utiliza una librería externa para facilitar las peticiones a la API REST, llamada Fast-Android-Networking [33], esto como vemos en la figura 5.8 hace que las llamadas sean realmente cortas y sin demasiada configuración.

```
public void getAllPOIs(JSONArrayRequestListener listener){
    AndroidNetworking.get(ServerConfig.getURLBase() +"/poi/")
        .setTag("poisRequest")
        .setPriority(Priority.LOW)
        .build()
        .getAsJSONArray(listener);
}
```

**Figura 5.8:** Ejemplo de una petición con la librería Fast-Android-Networking.

## 5.5. Resultados

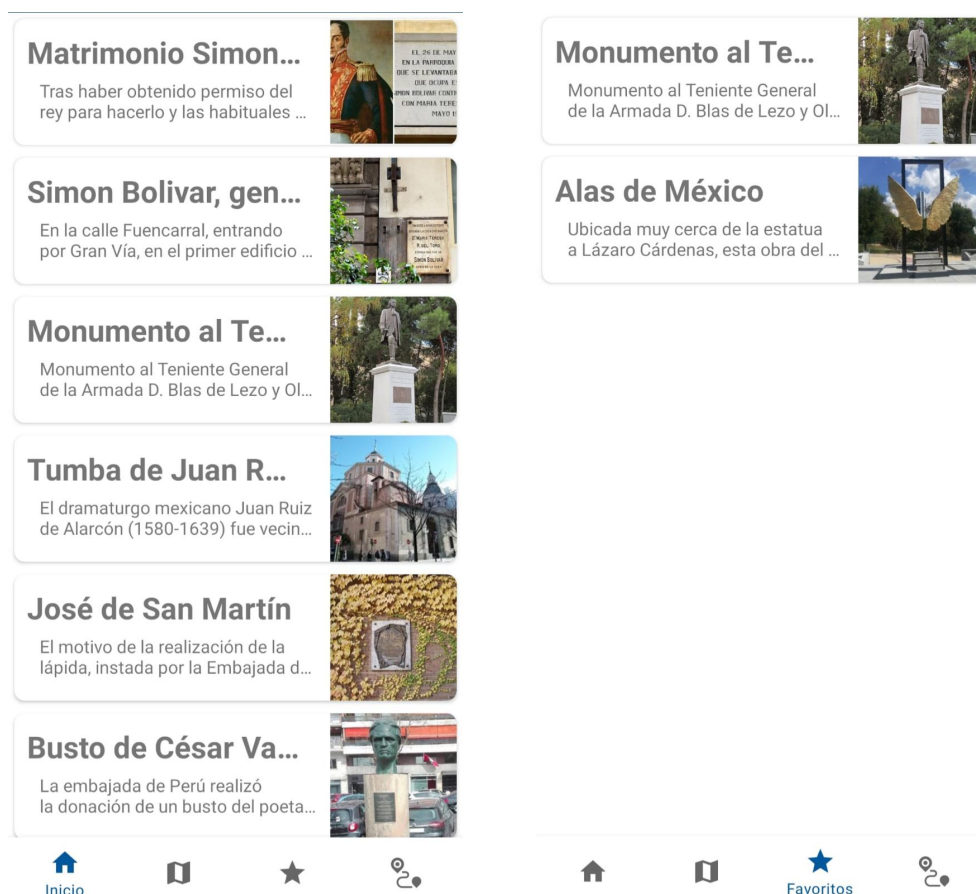
En esta sección se procede a explicar como se integra todo finalmente y como se mueven los datos desde la aplicación hasta la base de datos.

En la figura 5.10 se observa una llamada desde la aplicación en la que se piden todas los POI de una categoría, se ve entonces como desde el principio en la parte superior se crea la llamada, luego debajo de esta está el punto donde se recibe por parte del microservicio, aunque pasando primero por el gateway, el cual es transparente y no influye en la lógica de la petición, luego se accede a un servicio



que es el encargado de manejar la lógica y llamar al repositorio para acceder finalmente a la base de datos a través de la interfaz que **POIRepository**, además se comprueba como se utilizan los streams y lambdas de Java 8 para hacer el mapeo de un objeto a otro.

En la subfigura 5.9(a), se aprecia como la lista de objetos se representa después de recibir la respuesta del servidor, justo después de haberse realizado la petición mostrada en la figura 5.10; similarmente ocurre con los favoritos mostrados en la subfigura 5.9(b), pero en este caso se consultan los puntos preferidos por el usuario a la base de datos.



(a) Lista de puntos historicos en la aplicación MadLat

(b) Favoritos de la aplicación MadLat

**Figura 5.9:** Listas dentro de la aplicación MadLat

En subfigura 5.11(a) se ve la representación del perfil del punto, tras haber seleccionado un punto en alguna de las listas similares a las de las figuras 5.9, se observa así como están presentes las distintas opciones que se plantearon en el diseño en el capítulo 4, además en la misma figura 5.11 se ve la representación del mapa en la subfigura 5.11(b) donde se ven los sitios disponibles y la posibilidad de acceder a los perfiles pulsando sobre alguno de ellos.

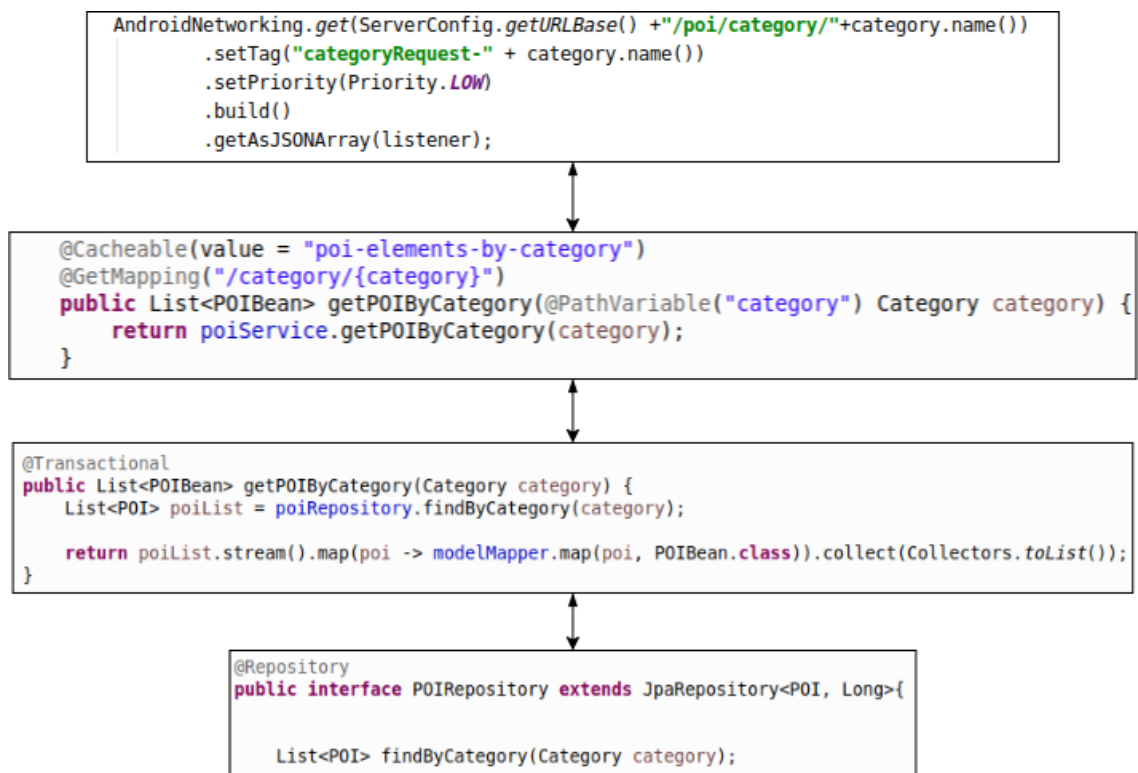
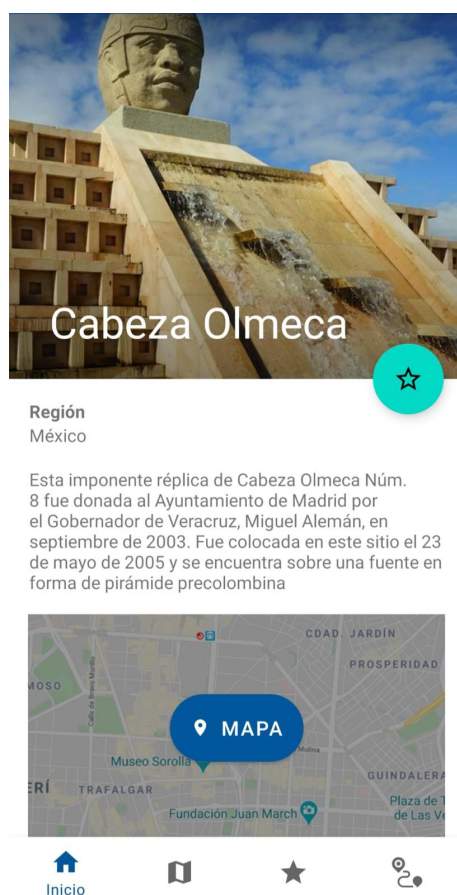
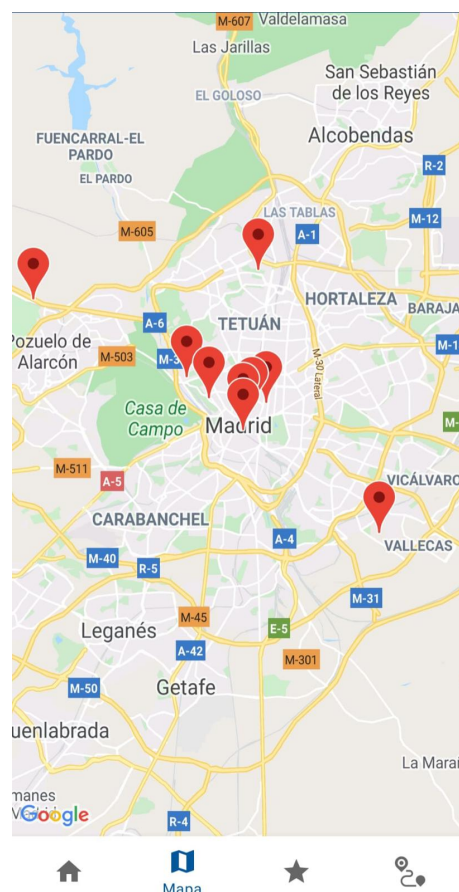


Figura 5.10: Ejemplo de una petición completa desde el front hasta la base de datos



(a) Perfil de un punto en la aplicación MadLat



(b) Mapa en la aplicación MadLat

**Figura 5.11:** Vistas de la aplicación



## CONCLUSIONES Y TRABAJO FUTURO

---

### 6.1. Conclusiones

En este Trabajo de Fin de Grado se ha realizado todo el proceso de análisis, diseño y desarrollo de una aplicación móvil para dar a conocer el patrimonio de América Latina en Madrid, se han llevado a cabo varias decisiones para que la aplicación fuese escalable en el tiempo y parte de lo ya se ha hecho pueda llevarse reutilizarse para otras plataformas y que esto pueda ser utilizado como una ampliación del trabajo ya realizado.

La idea partió de un proyecto del ayuntamiento de Madrid en colaboración con la Universidad Autónoma de Madrid, con el que se busca unir la comunidad latinoamericana con la comunidad madrileña, a razón de la gran historia interconectada entre ambas, esto, se ha conseguido en gran medida y supone un pequeño paso en la unión de la historia de estos pueblos.

Los aspectos técnicos logrados en la aplicación han provocado un gran aprendizaje en cuanto a como gestionar un proyecto por cuenta propia y en el descubrimiento de nuevas arquitecturas, como la basada en microservicios, además de como exponer un servidor correctamente para ser utilizado desde cualquier lugar del mundo y sin depender de servicios de terceros como puede ser el alojamiento en la nube de servidores o bases de datos en los que se paga por uso como puede ser, por ejemplo, Firebase.

Aunque este año 2020 ha tenido dificultades extraordinarias debido a la pandemia, también nos ha enseñado a gestionar desde casa todo lo necesario sin tener que desplazarnos, además de como manejar correctamente nuestro tiempo y aprender a tener paciencia en un mundo en el que todo iba demasiado rápido.

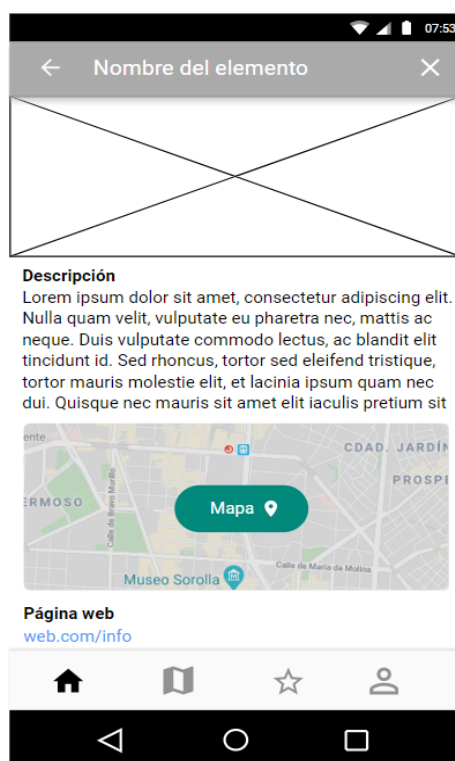
Con lo que con todo lo mencionado anteriormente este proyecto ha sido una experiencia enriquecedora, tanto tecnológicamente gracias a las herramientas empleadas como personalmente a nivel de organización.

## 6.2. Trabajo futuro

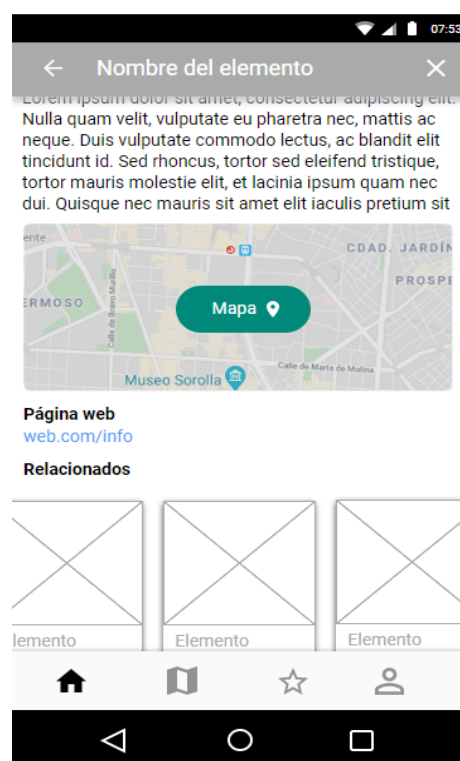
Después de una primera versión de la aplicación se plantean algunas mejoras y funcionalidades que pueden realizarse en la aplicación con el objetivo de complementar la experiencia:

- Soporte para iOS.
- Soporte para la Web.
- Valoración de los sitios y comentarios de los usuarios.
- Foto para el perfil del usuario.
- Que el usuario pueda valorar los comentarios más útiles para que aparezcan primeros.
- Poder acceder a audio que cuente sobre el punto de interés.

## ANEXO



(a) Vista inicial al entrar al elemento



(b) Vista inferior al bajar en la información del elemento

Figura 7.1: Información de un elemento.



Figura 7.2: Vista de favoritos de la aplicación

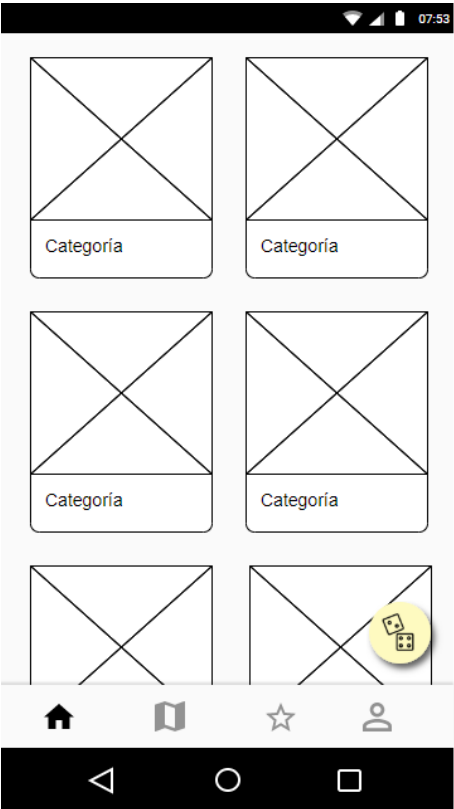
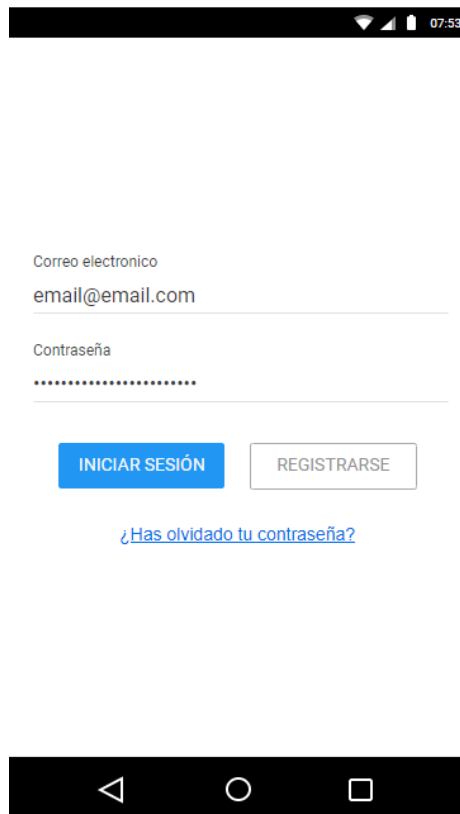
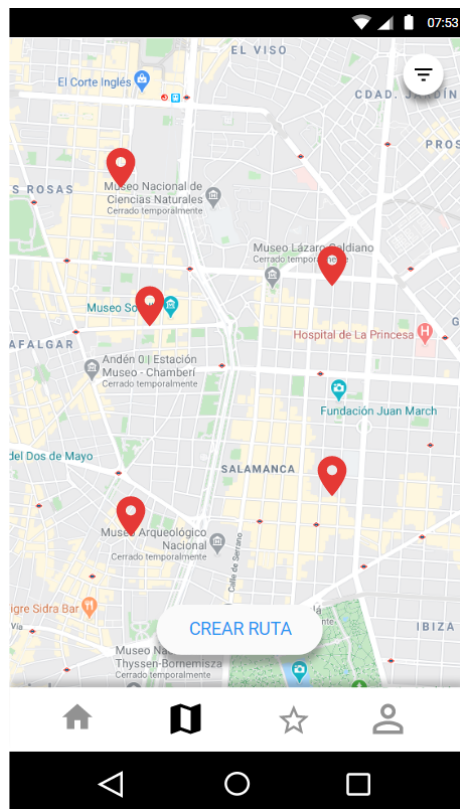


Figura 7.3: Vista inicial de la aplicación

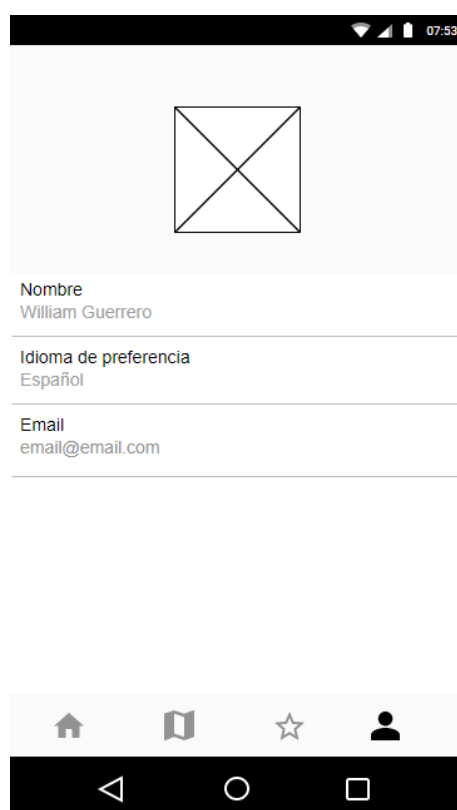




**Figura 7.4:** Vista del inicio de sesión de la aplicación



**Figura 7.5:** Vista del mapa en la aplicación



**Figura 7.6:** Vista del perfil del usuario en la aplicación

# BIBLIOGRAFÍA

---

- [1] "Aplicación aramedpi." <https://play.google.com/store/apps/details?id=es.arqueoweb.aramedpi&gl=ES>.
- [2] "Aplicación barcelona art culture." [https://play.google.com/store/apps/details?id=com.sutromedia.android.guide.bcn.unbound&hl=en\\_US](https://play.google.com/store/apps/details?id=com.sutromedia.android.guide.bcn.unbound&hl=en_US).
- [3] "Aplicación ciudades patrimonio de la humanidad." <https://play.google.com/store/apps/details?id=com.gvam.ciudadespatrimonio&gl=ES>.
- [4] "Aplicación world heritage." <https://play.google.com/store/apps/details?id=com.beacon.worldheritagesitespaid>.
- [5] "Aplicación culture trip." <https://play.google.com/store/apps/details?id=culturetrip.com>.
- [6] "Aplicación unesco heritage." <https://play.google.com/store/apps/details?id=fr.thebigtouffe.heritage>.
- [7] "What are microservices?." <https://microservices.io/>.
- [8] "Spring framework." <https://spring.io/projects/spring-framework>.
- [9] "Librería spring boot." <https://spring.io/projects/spring-boot>.
- [10] "Librería spring cloud." <https://spring.io/projects/spring-cloud>.
- [11] "Oauth2.0." <https://oauth.net/2/>.
- [12] "Oauth 2.0 security best current practice." <https://oauth.net/2/oauth-best-practice/>.
- [13] "Spring cloud gateway." <https://spring.io/projects/spring-cloud-gateway>.
- [14] "Service discovery." [https://cloud.spring.io/spring-cloud-netflix/multi/multi\\_\\_service\\_discovery\\_eureka\\_clients.html](https://cloud.spring.io/spring-cloud-netflix/multi/multi__service_discovery_eureka_clients.html).
- [15] P. Steve Burbeck, "Applications programming in smalltalk-80(tm): How to use model-view-controller (mvc)." <https://web.archive.org/web/20120729161926/http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>, 1992.
- [16] "Material design." <https://material.io/>.
- [17] "Android studio." <https://developer.android.com/studio>.
- [18] "Spring tool suite." <https://spring.io/tools>.
- [19] "Pgadmin." <https://www.pgadmin.org/>.
- [20] "postgresql." <https://www.postgresql.org/>.
- [21] "Android sdk." <https://developer.android.com/studio/releases/sdk-tools>.
- [22] "Qué es java?." [https://www.java.com/es/about/whatis\\_java.jsp](https://www.java.com/es/about/whatis_java.jsp).
- [23] "Herramienta para la creación de proyectos spring inicializ." <https://start.spring.io/>.

- [24] “Anotación `@springbootapplication`.” <https://docs.spring.io/spring-boot/docs/2.1.13.RELEASE/reference/html/using-boot-using-springbootapplication-annotation.html>.
- [25] “Anotación `@enableeurekaclient` para el servicio de descubrimiento.” [https://cloud.spring.io/spring-cloud-static/Dalston.SR5/multi/multi\\_\\_service\\_discovery\\_eureka\\_clients.html](https://cloud.spring.io/spring-cloud-static/Dalston.SR5/multi/multi__service_discovery_eureka_clients.html).
- [26] “`@enablecaching`, notación para habilitar la caché de spring.” [https://docs.spring.io/spring/docs/3.0.6.RELEASE\\_to\\_3.1.0.BUILD-SNAPSHOT/3.1.0.BUILD-SNAPSHOT/org/springframework/cache/annotation/EnableCaching.html](https://docs.spring.io/spring/docs/3.0.6.RELEASE_to_3.1.0.BUILD-SNAPSHOT/3.1.0.BUILD-SNAPSHOT/org/springframework/cache/annotation/EnableCaching.html).
- [27] “Librería spring data jpa.” <https://spring.io/projects/spring-data-jpa>.
- [28] “Dsl, creación de consultas a través de nombre de métodos.” <https://docs.spring.io/spring-data/data-jpa/docs/1.0.0.M1/reference/html/#jpa.query-methods.query-creation>.
- [29] “Java persistence api.” <https://www.oracle.com/java/technologies/persistence-jsp.html>.
- [30] “Api rest con spring.” <https://spring.io/guides/gs/rest-service/>.
- [31] “Spring beans.” <https://docs.spring.io/spring-javaconfig/docs/1.0.0.M4/reference/html/ch02s02.html>.
- [32] “Http methods.” <https://restfulapi.net/http-methods/>.
- [33] “Librería fast-android-networking.” <https://github.com/amitshekhariitbhu/Fast-Android-Networking>.



